TIM HARTNELL

OS SISTEMAS PERICIAIS EM MICROCOMPUTADORES

PRESENÇA

OS SISTEMAS PERICIAIS EM MICROCOMPUTADORES

TIM HARTNELL

OS SISTEMAS PERICIAIS EM MICROCOMPUTADORES

Inclui listagens para:

Spectrum/Spectrum +
Commodore 64
BBC Micro
Amstrad
Sistema MSX
Microsoft BASIC

Esta obra é dedicada a Bradley

INTRODUÇÃO

Benvindo a esta obra sobre o mundo fascinante dos sistemas «periciais». Nele tentaremos encontrar marcos específicos da história e desenvolvimento de alguns sistemas mais importantes, com o objectivo de compreender a situação actual destes sistemas periciais.

A partir disto o leitor poderá desenvolver sistemas próprios (incluindo um para detecção de falhas em automóveis, ou o MEDICI que lhe permitirá medir rapidamente o seu «stress» e prever o seu tempo aproximado de vida...). O principal sistema pericial apresentado neste livro é RITA. Este apresenta a estrutura genérica de um sistema de uso geral, que o leitor poderá modificar do modo que lhe parecer mais indicado. RITA é apresentado em acção seleccionando cães e gatos (!), usando as propriedades de um metal a fim de descobrir de que se trata, usando dados reais para prever o tempo em Inglaterra (e conseguindo resultados muito melhores do que aqueles que poderia obter por acaso).

Observaremos igualmente as linguagens que começam a dominar o mundo da Inteligência Artificial e dos sistemas periciais. Escrevi emuladores especiais da LISP e da PROLOG em BASIC, que o leitor poderá introduzir no seu computador para ter uma ideia de como funcionam estas linguagens. Apresenta-se primeiro uma linguagem nova e muito simples, a HASTE (de «HArtnell's Simple declarative TonguE!) que o ajudará a adaptar-se ao tipo de ambiente de programação em que se encontrará ao usar a LISP e a PROLOG.

Em resumo, esta obra foi pensada para ajudar o leitor a obter uma «base de conhecimento» razoável sobre o tema, suficiente para lhe permitir a construção dos sistemas periciais de que necessite.

TIM HARTNELL
LONDRES, 1985

1
O QUE É UM SISTEMA PERICIAL

Um sistema pericial é um programa de computador que contém os conhecimentos humanos sobre um tema, guardados de tal modo que os não peritos possam aceder, e utilizar, esse conhecimento. Um tal sistema é formado por duas partes — a informação armazenada e o sistema de raciocínio (que apresenta perguntas ao utilizador e toma decisões a partir das respostas dadas).

Os sistemas periciais podem fazer muito. Podem diagnosticar doenças infecciosas (MYCIN), deduzir estruturas moleculares a partir de espectrogramas de massa (DENDRAL), procurar petróleo e metais preciosos (PROSPECTOR) e ajudar-nos a fazer o carro arrancar de manhã (MECÂNICO, programa apresentado neste livro).

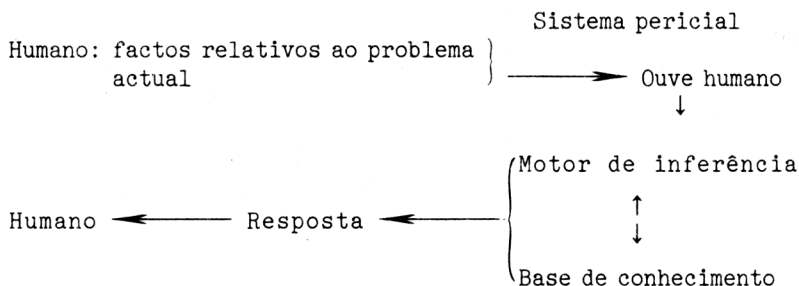
As características distintas de um sistema de base de conhecimentos inteligentes são as seguintes (e os termos empregues aqui deverão tornar-se fáceis de compreender depois da leitura):

1 — Especificidade a um domínio;
2 — Uso de um raciocínio «difuso» ou probabilístico;
3 — Disponibilidade de um modo de explicação das decisões tomadas;
4 — Separação clara entre os factos e o mecanismo de inferência;
5 — Possibilidade de crescimento por incrementos;
6 — Rentabilidade em termos financeiros;
7 — Tipicamente baseados em regras;
8 — Tendem a não ser excessivamente ambiciosos;
9 — A saída produzida é um conselho (por exemplo «mude as velas») em vez de consistir num novo grupo de dados (por exemplo um gráfico).

Para dizer a verdade, ainda não conhecemos verdadeiramente estes sistemas. É ainda muito cedo para dar uma resposta definitiva para esta pergunta aparentemente tão simples: o que é um sistema pericial? Não sabemos indicar quais as características, da lista já apresentada, que acabarão por ser as mais importantes, e quais se tornarão periféricas.

Até agora, o maior problema tem consistido em transformar os conhecimentos de um especialista para uma forma que a máquina possa manipular. O método actual, designado por «engenharia do conhecimento», está longe de ser eficiente, dado que obriga um especialista humano (o engenheiro de conhecimento) a extrair laboriosamente o conhecimento real a que um outro especialista humano recorre para resolver um problema. É esta a maior dificuldade na produção de sistemas periciais.

Os principais ingredientes

Sob a forma de um diagrama poderemos entender melhor as partes principais de um sistema pericial:

A «base de conhecimento», num sistema baseado em regras (como acontece no caso do MECÂNICO apresentado neste livro), é basicamente uma colecção de instruções IF/THEN como:

SE X E Y SÃO VERDADEIROS ENTÃO C É VERDADEIRO

O programa MECÂNICO contém linhas de codificação que significam por exemplo SE O MOTOR NÃO ARRANCA TALVEZ HAJA PROBLEMAS DE PLATINADOS. Começa por perguntar ao utilizador qual é o problema básico. Conduz depois o utilizador por uma árvore de possibilidades, percorrida em função das respostas deste. Este é um exemplo de «encadeamento» em que se procura uma solução única que está no fim de uma série de opções.

Outros sistemas periciais mais sofisticados empregam um encadeamento no sentido inverso (dito «backward», para trás, por oposição ao anterior, «forward»), em que se escolhe um objectivo (O CARRO NÃO TEM COMBUSTÍVEL) e se fazem perguntas a fim de estabelecer a verdade da hipótese em causa.

A base de conhecimentos

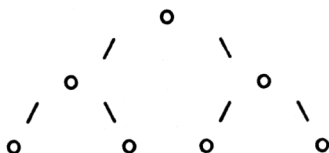
Esta pode estar directamente codificada no corpo principal do programa (como acontece em MEDICI ou em MECÂNICO), caso em que é mais ou menos impossível eliminá-la, ou encontrar-se numa base de dados endereçável, de tal modo que pode ser acedida, modificada e actualizada mesmo quando o programa está a correr. RITA constrói um conjunto de regras que determinam a importância atribuída a diversas entradas, dando valores a variáveis (e alterando estes valores à luz das tentativas ulteriores e da resposta do utilizador).

No caso de um programa do primeiro tipo (e contrastando com o programa de tipo RITA, que produz as suas próprias regras quando está em funcionamento, mesmo que o utilizador não tenha qualquer ideia sobre quais possam ser essas regras), os conhecimentos devem ser adquiridos directamente a partir de um especialista humano.

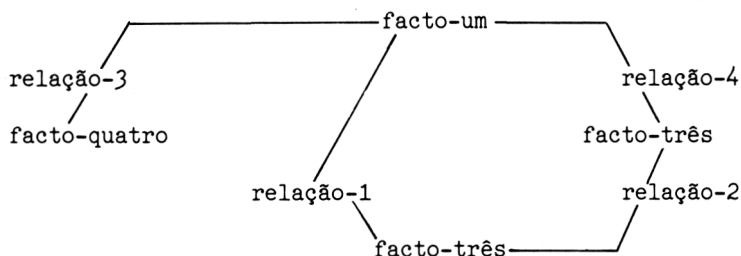
Um método que permitisse a um computador obter directamente o conhecimento, criando assim a sua própria base de trabalho, seria talvez o mais importante desenvolvimento da história dos sistemas periciais. Os conhecimentos deste tipo, obtidos directamente pela máquina, constituiriam evidentemente um recurso precioso. Aqueles que satisfizerem esta necessidade receberão uma fortuna (a captura de conhecimentos a partir de especialistas humanos é discutida em algum pormenor no capítulo sete, «Knowledge Engineering», do livro *Rule-Based Expert Systems* (B. Buchanan, 1984)).

Estão agora a decorrer vários projectos de investigação, em vários pontos do mundo, no campo da «aprendizagem pela máquina», a tentativa de levar um computador a descobrir e codificar os conhecimentos humanos por si mesmo.

Vejamos algumas das estruturas de raciocínio actualmente usadas pelos sistemas periciais.

1 — ÁRVORES DECISIONAIS. O computador segue um trajecto por uma série de perguntas, deixando que a resposta a cada uma delas determine qual o canal específico que o programa deve percorrer. Este tipo de programa necessita de dados internos. Aprecia situações do tipo «Sim/Não». Não é capaz de tratar facilmente situações do tipo «difuso», do género «geralmente/talvez/por vezes».

Certos programas disponíveis comercialmente, como CONSULT e EXPERT-EASE utilizam árvores decisionais.

2 — REGRAS DE PRODUÇÃO. Esta estrutura é observada em programas comerciais como o XCON e o DART, que utilizam processos de tipo IF/THEN (juntamente com AND).

3 — REDES SEMÂNTICAS. O MYCIN e o PROSPECTOR são exemplos deste tipo de estrutura:

Os sistemas periciais e a inteligência artificial (AI) são temas quentes neste momento. Na edição de Janeiro de 1985 da revista *Byte* (págs. 257-282), sob o título «Expert Systems — Myth or Reality?», Bruce d'Ambrosio indica que o Departamento de Defesa americano apresentou a Inteligência Artificial como uma das 10 tecnologias mais importantes a desenvolver nos últimos anos deste século. A «Japan Inc.» entrou já em acção com o seu «projecto de computador de quinta geração». Um objectivo primordial deste projecto é o desenvolvimento de melhores sistemas periciais.

Os principais temas em investigação e desenvolvimento para os computadores de quinta geração (de acordo com as actas da Conferência Internacional sobre Sistemas de Computadores de Quinta Geração, 19-22 de Outubro, 1981) incluem o seguinte:

SISTEMAS DE APLICAÇÃO BÁSICOS	Tradução por máquina Resposta a perguntas Compreensão aplicada de voz Compreensão aplicada de imagens Solução de problemas aplicada
SISTEMAS DO SOFTWARE BÁSICOS	Gestão de bases de dados Resolução de problemas e inferência Interface inteligente

ARQUITECTURAS AVANÇADAS	Máquina inteligente Máquina de programação lógica Máquina funcional Máquina de álgebra relacional Máquina de suporte de tipos de dados abstractos Máquina de fluxo de dados
ARQUITECTURAS DE REDES/ /FUNÇÕES DISTRIBUÍDAS	Máquina de bases de dados Máquina de computação numérica de alta velocidade Sistema de comunicação homem--máquina de alto nível
TECNOLOGIA VLSI	Arquitectura VLSI (integrada em escala muito grande) Arquitectura CAD VLSI inteligente (concepção apoiada por computador)
TECNOLOGIA DE SISTEMAS	Sistema de programação inteligente Sistema de concepção de bases de conhecimentos

Incitadas pelos esforços desenvolvidos neste campo pelo Japão, tanto a França como a Inglaterra estão agora a estudar projectos nacionais (talvez também o leitor possa contribuir para a investigação no seu país partindo do que aprender neste livro...). No capítulo que se segue, veremos como se podem usar na prática algumas das ideias já esboçadas.

2

SISTEMAS QUE FUNCIONAM

Os sistemas periciais melhor conhecidos actualmente em uso são:

— MYCIN, trabalhando no campo das doenças infecciosas;
— DENDRAL, que analisa compostos químicos orgânicos;
— PROSPECTOR, que sugere locais para exploração mineralógica.

Neste capítulo observaremos estas três aplicações. Podemos tirar delas algumas lições gerais. Como MYCIN parece ser a mais importante das três, começaremos por ela.

As experiências MYCIN em Stanford

Começaremos pelo sistema MYCIN, desenvolvido em Stanford, muito útil no diagnóstico de doenças infecciosas. O MICYN é um dos sistemas com mais êxito actualmente em uso, e podemos aprender muito estudando-o. A partir das informações dadas nesta secção o leitor compreenderá talvez a forma como pode desenvolver e expandir sistemas próprios.

O MYCIN é um dos mais antigos sistemas periciais em uso. Trabalha baseando-se em regras, numa área em que não se pensava anteriormente que fosse possível codificar dados envolvidos em tomadas de decisões.

O MYCIN foi precedido pelo DENDRAL no Stanford Heuristic Programming Project, sendo este último capaz de deduzir estruturas moleculares a partir da análise das saídas de um espectrógrafo de massa. No entanto, o DENDRAL não foi concebido desde início como sistema pericial; evoluíu nesse sentido a partir de uma investigação sobre a indução científica. Por outro lado, o DENDRAL trabalha num campo em que os processos envolvidos na tomada de decisões eram relativamente bem conhecidos antes de o sistema ser construído. O MICYN começou como um projecto de construção de um sistema pericial, num campo em que as regras de tomada de decisões estavam mal definidas, e o projecto teve êxito (estudaremos o DENDRAL em algum pormenor mais adiante).

O MYCIN não foi concebido para substituir um médico, mas sim para actuar como conselheiro. O doente é examinado por um médico, que indica ao MYCIN os sintomas observados. O sistema apresenta então uma sugestão de diagnóstico, muito semelhante à que um especialista daria nas mesmas circunstâncias. Além de dar sugestões, o MYCIN pode explicar como chegou à conclusão enunciada. Um desenvolvimento deste aspecto do comportamento do MYCIN é um sistema pericial que ensina estudantes de medicina, criando assim novos peritos humanos a partir dos conhecimentos dos antigos, sob forma codificada (a obra de Buchanan já citada é um bom texto de consulta sobre o projecto MYCIN; será este o livro a estudar se o leitor estiver interessado pela evolução do projecto, e pelas concepções que lhe estão na base).

Um outro desenvolvimento do MYCIN, o EMYCIN (significando MYCIN «essencial»), foi construído a partir daquele retirando-lhe os conhecimentos de diagnóstico codificados (a «base de conhecimentos») e mantendo o mecanismo de raciocínio («o motor de inferência»). Isto permitiu a criação de sistemas periciais noutros temas, mais ou menos simplesmente, introduzindo novas bases de conhecimento no EMYCIN. Este facto sugere que há lições a tirar da estrutura do EMYCIN que podem ser aplicadas em geral aos sistemas periciais baseados em regras.

O «Gang» MYCIN

A equipa que desenvolveu o MYCIN (e que acabou por ser conhecida pelo «gang» MYCIN) começou a trabalhar neste sistema pensando seguir a ideia de um programa chamado SCHOLAR, capaz de «falar» sobre a geografia da América do Sul. No entanto, verificou-se que os dados médicos não cabiam em categorias tão estritas como os factos geográficos, pelo que a linha SCHOLAR foi abandonada.

A experiência mostrou que os desenvolvimentos mais frutuosos ocorrem no sentido da codificação explícita de regras específicas — como as usadas pelos especializados humanos — que ligam elementos da base de dados, em vez de tentar dividir estas regras, de modo a criar trajectos genéricos que o processo de tomada de decisões possa seguir. A disponibilidade deste tipo de regras significa ainda que o MYCIN pode explicar como chega às suas conclusões, usando o tipo de declarações que os próprios médicos poderiam utilizar (entre as pessoas envolvidas no desenvolvimento do MYCIN citamos J. Aikins, S. Axline, J. Bennett, A. Bonnet, B. Buchanan, W. Clancey, S. Cohen, R. Davis, L. Fagan, F. Rhame, C. Scott, E. Shortliffe, W. vanMelle, S. Wraith e V. Yu).

As regras

O primeiro problema encontrado, depois de se ter definido a natureza da base de dados, foi o facto de os conhecimentos humanos so-

bre doenças infecciosas não se encontrarem claramente definidos em grupos que se excluíssem mutuamente. A barreira entre cada dois elementos deste domínio era geralmente difusa.

A primeira versão do programa possuía 200 regras, apresentadas no essencial como se fossem instruções IF/THEN (e incluindo por vezes a ELSE). Nas versões ulteriores, as ELSE's foram eliminadas, dado que raramente eram usadas. O MYCIN possui hoje mais de 600 regras.

Decidiu-se que o MYCIN deveria tormar decisões da seguinte ordem:

— Determinar se a infecção se deve a algum organismo conhecido;
— Definir qual o organismo;
— Decidir quais as drogas a utilizar para contrariar a infecção;
— Sugerir o melhor tratamento geral.

```
Se: (1) a aparência do organismo é gram-positiva, e
    (2) a morfologia do organismo é coco
    (3) a formação de crescimento do organismo é ramificada,
então há justificação (0.7) para considerar que
  a identidade do organismo é estafilococo
```

Note-se o número (0.7), que indica o grau de certeza atribuída pelo MYCIN à sua própria conclusão.

O grau de certeza é expresso numa escala que varia entre –1 e + 1, onde –1 é «falso» (negação aboluta de uma hipótese), 0 significa que não existem provas a favor ou contra a hipótese e 1 equivale a considerar a hipótese demonstrada. Aqueles que conceberam este sistema consideram-no imperfeito, mas como tendo dado bons resultados. Indicam que deve apenas ser considerado como um ponto de partida para a construção, de um modo lógico e coerente, de tomada de decisões em áreas onde os dados são incertos, tal como o domínio onde se aplicam (Shortliffe, 1983).

A regra é impressa para consulta do médico na forma clara acima indicada. No entanto, as regras são guardadas internamente sob a forma de listas LISP, do seguinte modo:

```
PREMISSA: ($AND (MESMO CNTXT GRAM GRAMPOS)
                (MESMO XNTXT MORF)
                (MESMO CNTXT CONFORM RAMIFICADA))
ACÇÃO: (CONCLUIR CNTXT IDENT ESTAFILOCOCO PROB .7)
```

(Esta lista terá provavelmente mais sentido para o leitor depois de ler a secção deste livro dedicada a EASLE e à LIPS-A).

Um sistema pericial será muitas vezes usado por pessoas que têm conhecimentos inferiores aos incorporados no sistema. Isto significa que o raciocínio subjacente às conclusões a que o programa chega

nem sempre será óbvio. O MYCIN possui um subprograma chamado CHRONICLER que pode responder a perguntas relativas à terapia advogada. Um diálogo com este programa terá a seguinte aparência.

** PORQUE DEU A DROGA X PARA A SITUAÇÃO Y?
X foi receitado para Y
Porque
— X é muitas vezes usado para Y na doença Z
— o elemento Q dos dados do doente mostra intolerância à droga P
— X não é contra-indicado.

A engenharia do conhecimento é um processo lento e caro. Para conseguir que o MYCIN melhorasse a sua base de dados ganhando conhecimentos e eliminando os erros, incoerências e omissões, foi desenvolvido um subprograma chamado TEIRESIAS. Quando um médico encontra um erro, ou um diagnóstico incompleto, pode accionar TEIRESIAS, que lhe fará algumas perguntas a partir das quais formará uma nova regra.

O MYCIN foi concebido para comunicar com o médico que o utiliza da forma mais natural possível, e este facto observa-se claramente numa conversação com TEIRESIAS. A saída do computador durante a formação de regras inclui linhas «humanas» como:

```
Desagrada-me criticá-lo, Dr..., mas sabe que
a maior parte das regras nesta área incluem...
Deverei tentar escrever uma cláusula que cubra...
```

Depois de a nova regra ter sido acrescentada à base de conhecimentos, é verificada pelo próprio médico, que é guiado pela máquina em todo o processo de verificação:

```
Foi acrescentada a regra xyz à base de conhecimentos
Vou reexecutar a consulta a fim de verificar a
eficácia da nova regra. Sente-se confortavelmente
porque vou demorar um pouco.
```

Depois de as regras serem introduzidas, testadas e corrigidas, o MYCIN compila-as para uma estrutura em árvore, depois passada a código-máquina.

EMYCIN

O EMYCIN possui dois componentes principais, a base de conhecimentos e o motor de inferência. Em 1974, dois anos depois de se iniciar o projecto MYCIN, estavam a realizar-se as primeiras experiên-

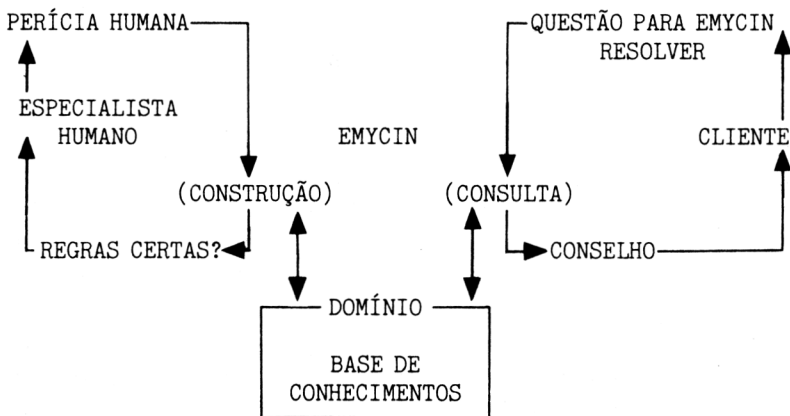
cias de substituição da base de conhecimentos de diagnóstico por outras, noutros campos. Usou-se um manual de um automóvel Pontiac como fonte de conhecimentos que desse ao motor uma inferência material do qual pudesse partir. Foram escritas quinze regras relativas ao circuito da buzina (se bem que a sua estrutura seja bastante diferente, a natureza dos conhecimentos encerrados no programa MECÂNICO deste livro assemelha-se às regras dadas a este primeiro «filho» de MYCIN).

A experiência ganha no campo do Pontiac, e outras experiências do mesmo tipo, conduziram ao EMICYN. Este é formado por uma «concha» constituída pelo motor de inferência e por uma porção significativa do sistema TEIRESIAS (que, como o leitor certamente recorda, é usado para acrescentar novas regras à base de conhecimentos através de uma conversação directa com um utilizador).

Se se executar EMYCIN para criar um novo sistema pericial, o programa conduz-nos passo a passo há criação do sistema, ao longo de um diálogo como o que se segue:

```
> Deseja criar uma nova base de conhecimentos?
> Escreva uma palavra ou frase que descreva o domínio:
> Escreva um nome de uma palavra para a raiz da sua árvore de con-
  texto, o «objecto» central a que a consulta respeita
```

Depois de a informação ter sido definida, o EMYCIN passa a ser especialista nesse domínio, e pode ser usado como tal. O programa possui portanto duas funções perfeitamente separadas. A primeira consiste em aceitar dados e criar as regras que lhe estão adequadas, e a segunda em fazer deduções a partir destas regras.

Vejamos as funções do EMYCIN sob a forma de um diagrama:

Um dos principais problemas na criação de novos sistemas periciais a partir do EMICYN é a falta de precisão de muitas afirmações em linguagem natural. É certo que esta falta de precisão pode ser quantificada até certo ponto na construção de regras (como vimos no MYCIN), mas nem sempre é possível fazê-lo, especialmente quando EMYCIN pode estar a interpretar incorrectamente os dados introduzidos. James Bennett (Buchanan, 1984) está actualmente a desenvolver um sistema especializado na criação de outros (designado por ROGET), o qual auxiliará o trabalho com o EMYCIN, conduzindo a uma resposta mais eficaz dos «filhos» de MYCIN.

Já foram desenvolvidos a partir deste sistema alguns outros bastante eficazes. Incluem-se entre eles o SACON, o CLOT e o PUFF (Buchanan, 1984). O SACON procura definir uma estratégia de análise para um dado problema envolvendo uma estrutura específica (onde o sistema infere comportamentos materiais como uma deflexão excessiva, etc.), enquanto o CLOT produz sugestões sobre problemas de hemorragia. Um outro programa especializado em problemas médicos, o PUFF, produz sugestões sobre o diagnóstico de doenças pulmonares, enquanto dois descendentes deste, o CENTAUR e o WHEEZE utilizam a base de conhecimentos do PUFF mas manipulam-na usando diferentes representações e estruturas de controlo. Um outro programa, VM (de Gestor de Ventilação) vigia o estado pós-operatório de um doente após operações cardiovasculares, quando é muitas vezes necessária uma assistência mecânica à respiração.

DENDRAL, um feiticeiro químico

O DENDRAL, que foi escrito antes de MYCIN, é um programa que analisa compostos orgânicos a fim de determinar a sua estrutura. Este sistema pericial, desenvolvido por E.A. Feigenbaum e J. Lederberg em Stanford, é tão eficaz na sua tarefa que se pensa que nenhum químico humano pode obter os mesmos resultados. Algumas das análises do DENDRAL foram já publicadas como resultados originais de investigação (Rich, 1983). Descobriu igualmente erros em tabelas químicas publicadas (Raphaele, 1976).

O programa tenta inferir a composição química e a estrutura orgânica dos composto químicos, usando dados de um espectrograma de massas da substância, e leituras de ressonância magnética nuclear.

O DENDRAL segue basicamente um processo em três passos ao tentar determinar a estrutura mais provável do composto. Esses três passos são o planeamento, geração de respostas e teste (Buchanan, 1981).

Na fase de planeamento, o sistema utiliza as regras nele incorporadas para limitar o tipo de estruturas que serão testadas, eliminando

as incompatíveis com os dados recebidos. Na segunda fase são produzidas respostas possíveis, de acordo com a base de regras do sistema. A fase final consiste em testar esta estrutura. Sem a fase de planeamento, seriam eventualmente geradas milhões de estruturas possíveis, levando o sistema a fornecer respostas inaceitavelmente lentas. Nesta fase, as possibilidades são investigadas de modo bastante meticuloso, a fim de garantir que existem sólidas razões químicas para a eliminação de qualquer das estruturas.

O DENDRAL pode localizar os elementos mais importantes dos dados de teste, e utilizá-los juntamente com o seu conhecimento básico da forma como as estruturas químicas são constituídas. Testa as suas descobertas recorrendo a um «espectrómetro de massa simulado» interno, o qual verifica se a resposta potencial produz o mesmo espectrograma que a substância testada. Tal como o MYCIN, o DENDRAL utiliza um conjunto de regras empíricas que lhe foram fornecidas por especialistas humanos.

As leituras do espectrómetro de massa constituem um dos tipos de dados mais importantes utilizados pelo DENDRAL. Um espectrómetro de massa separa os átomos e moléculas de acordo com as respectivas massas, carregando-os electricamente de tal modo que possam ser usados campos magnéticos ou eléctricos para efectuar a separação. O espectrómetro produz um gráfico em que as alturas relativas dos picos estão relacionadas com as populações comparativas de certos iões da amostra.

O DENDRAL analisa o espectrograma de uma substância a identificar, e utiliza as suas regras para produzir uma lista de subestruturas que devem existir na substância desconhecida e outra das que não devem existir.

Observemos uma regra DENDRAL (número 75; Winston, 1984). Nota-se que é da mesma «família» das regras MYCIN observadas mais atrás:

Se existe um pico positivo a 71 unidades de massa atómica
existe um pico positivo a 43 unidades de massa atómica
existe um pico positivo a 86 unidades de massa atómica
existe qualquer pico a 58 unidades de massa atómica
então deve existir uma estrutura N-PROPIL-CETONA3

O mecanismo de produção de hipóteses do DENDRAL é chamado CONGEN, e recebe entradas sob a forma de leituras do espectrómetro de massa e outros testes químicos. Por outro lado, o gerador permite a imposição de limites às estruturas que produz, como o número de átomos de cada tipo presentes na molécula e quaisquer outros constrangimentos determinados pelo operador (Buchanan, 1981). A partir deles, o CONGEN produz uma lista completa de todas as estruturas aceitáveis em função dos dados introduzidos.

Se bem que o DENDRAL não possa ser utilizado por quem não estiver treinado nas áreas específicas da química que correspondem ao seu domínio, podendo-se portanto considerar que fracassa num dos critérios de definição dos sistemas periciais (o colocar os seus conhecimentos à disposição de não especialistas), funciona melhor do que estes (Simon, 1984). Forneceu resultados valiosos numa vasta gama de casos, incluindo (segundo Buchanan, 1981) a determinação da estrutura química de compostos desconhecidos nas seguintes áreas:

— ácidos orgânicos em fluídos
— impurezas em produtos químicos manufacturados
— antibióticos
— hormonas de insectos

Sistemas milionários

O PROSPECTOR parece um sonho tornado realidade. Muitos relatos das suas acções sugerem que basta descrever ao programa uma determinada área, e que ele responde o que se deve procurar, petróleo, ouro ou diamantes. Mas, como é evidente, a realidade (se bem que excitante) não é tão mágica como alguma imprensa sugeriu.

Apesar disto, parte do trabalho de PROSPECTOR tem sido bastante valioso. Por exemplo, este sistema pericial foi muito importante na descoberta de um depósito de molibdénio perto do Monte Tolman, em Washington (Winston, 1984). Neste caso, os especialistas humanos discordaram da opinião de PROSPECTOR e só a escavação demonstrou que o sistema tinha razão.

A base de conhecimento de PROSPECTOR é formada por mais de 500 regras e 300 afirmações. As suas regras de inferência funcionam de modo semelhante às regras de produção do MYCIN (Simons, 1984). No entanto, enquanto MYCIN pode combinar várias possibilidades (um doente pode ter mais do que uma infecção simultaneamente), o PROSTECTOR utiliza inferências probabilísticas estritas (ver o apêndice A deste livro, sobre o teorema de Bayes), partindo do princípio de que não podem ocorrer simultaneamente dois resultados (Rich, 1983).

O conhecimento guardado por PROSPECTOR foi obtido da forma habitual, e trabalhosa, que consiste em perguntar a técnicos humanos (geólogos, neste caso) a forma como trabalham, e em codificar as suas respostas. É muito interessante que cada um dos especialistas humanos interrogados tenha afirmado que o processo de engenharia do conhecimento lhes permitiu desenvolver as suas próprias capacidades neste campo, dado que se viram forçados a tomar consciência do que fazem, e de como o fazem (Duda, 1981).

3

CRIAÇÃO DE SISTEMAS

Há vários passos necessários no desenvolvimento de um sistema pericial. Começaremos pelo mais óbvio, que consiste em definir o problema. Isto pode ser feito em termos exactos ou usando a linguagem natural. A partir deste ponto, teremos de descrever a natureza das entradas de dados que serão usadas para chegar a uma conclusão (a saída do sistema pericial).

Usando esta informação, é possível determinar os mecanismos teóricos que permitirão a resolução do problema (os «cálculos simbólicos que serão realizados).

Chegamos agora à parte difícil, a saber, a transformação da percepção do problema, e o nosso desejo de seguir um determinado trajecto de raciocínio para encontrar a solução, num procedimento (ou série de procedimentos) que dará resultados práticos. É útil, nesta fase, considerar a incorporação de um «mecanismo de relatório» no sistema, o qual indicará ao observador humano a razão porque foram seguidos certos trajectos e, em última análise, explicará o porquê da conclusão final do sistema. Os procedimentos deverão idealmente permitir a auto-modificação dos seus elementos, o que garante que os trajectos mais úteis são executados rapidamente, poupando tempo e energias para as questões que possam ser mais difíceis na análise.

As duas condições fundamentais, e razoavelmente óbvias, que deverão ser preenchidas para que um sistema pericial seja útil são as seguintes. A primeira é que o sistema deve fazer aquilo que diz. Se se constrói um sistema para distinguir entre diversos tipos de conchas marinhas, deve ser capaz de exibir uma capacidade de raciocínio adequada quando confrontado com a vasta maioria dos objectos pertencentes a este domínio. Em segundo lugar, por muito «esperto» que o sistema seja deste ponto de vista, é improvável que seja usado a menos que permita uma interacção simples com o utilizador.

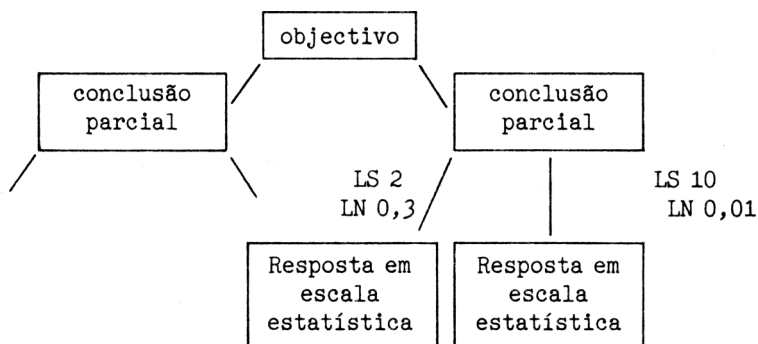
Se bem que seja muito fácil construir, por exemplo, um programa que trate a contabilidade pessoal, o uso de um sistema pericial exige mais do que um simples conhecimento do formato das entradas de dados e das saídas pretendidas. Um sistema pericial é um dispositivo complexo, mas limitado. Para ser usado convenientemente, é necessário que o utilizador compreenda os limites em que o programa é útil.

É provável que um utilizador necessite de gastar muito tempo com o programa antes de começar verdadeiramente a compreender as suas possibilidades e restrições. A apresentação no visor, e a documentação do programa devem contribuir tanto quanto possível para diminuir o tempo gasto até o programa poder ser usado de uma forma positiva.

Tipos de informação

Como é óbvio, será conhecido o domínio geral do programa («deficiências vitamínicas»), e o contexto das entradas («pele seca», etc.) e das saídas («falta de vitamina A») será razoavelmente fácil de definir na maior parte dos sistemas. No entanto, o tipo de informação que o programa pode tratar demorará algum tempo a definir.

Por vezes, quando se desenvolvem sistemas periciais, é bastante útil poder indicar ao sistema que algumas respostas são mais importantes do que outras. Imagine-se um programa de diagnóstico médico dedicado a problemas nas costas. Uma resposta a uma pergunta como «O doente sofre de dores quando segura um peso?» pode ser mais importante do que a resposta à pergunta «O doente come vegetais todos os dias?». Neste caso, o sistema pericial necessita de um método que lhe permita sublinhar as resposta mais importantes, e noutos casos diminuir o valor de respostas.

Para isto, usa-se aquilo que é designado por valores de *suficiência lógica* para as respostas cuja importância deve ser sublinhada, e valores de *necessidade lógica* para as menos importantes. Para os utilizar, multiplica-se uma resposta pelo valor de suficiência lógica se for verdadeira, e divide-se pelo valor de necessidade lógica se for falsa. Isto garante que o sistema tende a privilegiar a resposta que interessa, em vez de a deixar submersa num grupo de dez respostas pouco relevantes.

Imaginemos que desenvolvemos um sistema dedicado ao diagnóstico médico que segue uma árvore decisional, com perguntas e regras nos diversos modos. Este tipo de sistema pericial pode partir de instruções de objectivo, fazer as perguntas apropriadas, e usar em seguida a sua base de regras para tomar decisões.

O nosso sistema está preparado para fazer as perguntas apropriadas («O doente sua profusamente?»), aceitar a resposta escolhida por entre um menú de possibilidades («SIM» a «NÃO SEI» e «NÃO»), converter esta resposta numa probabilidade, multiplicar em seguida as respostas verdadeiras pelo valor de suficiência lógica correspondente a essa pergunta (LS no esquema anterior) e dividir as respostas falsas pelo valor de necessidade lógica que lhes corresponde (LN no esquema anterior).

É muitas vezes desejável que um sistema pericial contenha um mecanismo que lhe permita interromper certas linhas de interrogatório se forem dadas respostas que indicam que estas não têm importância. Depois de «O MOTOR CAÍU DO CARRO?» (que poderá ter uma resposta «SIM»), não valerá a pena continuar a analisar o raciocínio que conduz à resposta «TENTE LIGAR O MOTOR DE ARRANQUE; O MOTOR FUNCIONA?»; esta pergunta produzirá respostas que não podem ser avaliadas para tirar conclusões, e demonstra de modo convincente a falta de inteligência que o sistema de facto tem.

Além de um modo de evitar que certas perguntas inúteis venham a ser feitas mais tarde, necessitamos de fazer certas perguntas quando forem dadas respostas positivas a perguntas anteriores, e só nesse caso.

Dois tipos de raciocínio

Parecem existir duas categorias gerais em que podem ser incluídas as actividades de raciocínio. O conhecimento destas categorias torna mais simples criar sistemas periciais, desde que seja possível (e é-o quase sempre) decidir qual a categoria em que o tipo de problemas, a resolver pelo sistema, pode caber.

À falta de melhores descrições, prefiro chamar a estas categorias raciocínio «exacto» e «aproximado». Diz-se que um problema lógico é exacto quando existe — para todas as entradas — uma saída «verdadeira». Um problema aproximado será aquele em que a verdade das entradas é avaliada numericamente, sendo a saída qualificada do mesmo modo.

Como sabemos, as duas partes principais de muitos sistemas periciais (como o programa de diagnóstico de doenças infecciosas, o MYCIN, observado no capítulo anterior) são a base de conhecimentos (a experiência guardada na máquina) e o motor de inferência (o me-

canismo de raciocínio). Vimos que grande parte da informação que temos sobre o mundo não pode ser reduzida a números. As nossas vidas estão preenchidas por situações em que usamos frases como «parece provável», «muitas vezes», «por vezes», «sinto que», etc. Para que os sistemas periciais possam ter qualquer uso eficaz no mundo não ideal nem rigoroso onde vivemos, é necessário que sejam capazes de trabalhar usando os conceitos e a percepção humanos.

Os seres humanos não raciocinam apenas (ou sequer principalmente) seguindo regras. Em muitos casos, o peso dado a certos elementos no interior da proposição lógica baseia-se num «conhecimento de experiência feito». Isto é demonstrado claramente na compreensão de texto escrito ou falado. As palavras de um texto só têm significado num contexto bem definido. A extensão dos nossos conhecimentos sobre o ambiente em que o texto ocorre, e que reflecte e modela, ditam o grau de compreensão manifestado relativamente ao texto. Esta simples afirmação sugere já a dificuldade com que deparamos no desenvolvimento de sistemas periciais que interactuem connosco em linguagem natural. A tecnologia actual só nos permite instalar uma pequena fracção da riqueza dos nossos conhecimentos na máquina. E sem esta riqueza, a compreensão da linguagem natural só pode ser eficaz (e aliás pouco) dentro de limites bastante estreitos.

4

SISTEMAS BASEADOS EM REGRAS

Pode-se criar facilmente um sistema pericial primitivo baseado em regras. Um tal sistema, talvez o tipo mais simples de sistema pericial que pode ser escrito, pode no entanto ser eficaz e útil, além de dar uma ideia de como foram criados sistemas mais complexos baseados em regras, como o programa MYCIN para doenças infecciosas.

Neste capítulo introduzirei o programa MECÂNICO, um sistema simples baseado em regras que é concebido para ajudar o automobilista ignorante em mecânica (como eu próprio) a determinar possíveis causas dos problemas que surgem nos seus carros. Como este programa se baseia em informação profissional extraída com algum esforço de especialistas humanos neste campo, espero que possa ser útil ao leitor.

No entanto, a principal razão para o incluir neste livro consiste em mostrar como pode ser desenvolvido um sistema baseado em regras, no qual estas se encontram codificadas por uma ordem bem definida, de tal modo que a resposta a uma pergunta determina qual a pergunta que o sistema fará a seguir.

Duas faces de cada regra

A regra mais simples usada num sistema pericial possui duas partes. Um lado «esquerdo» (LHS), que consiste tipicamente numa instrução IF («SE o computador se incendiou»...) e um lado direito (RHS) que consiste geralmente numa cláusula THEN («ENTÃO ponha-o na banheira»). Um sistema pericial baseado em regras pode consistir em pouco mais do que uma série de pares de LHS-RHS apresentados por uma ordem específica.

Um sistema primitivo deste tipo permitirá ao utilizador escrever a resposta a uma pergunta (como «Está o computador a arder serenamente ou a transformar-se numa horrível bolha de plástico?»), procurando depois na sua colecção de regras uma possível concordância entre o LHS de uma regra e a resposta dada. Neste ponto imprimirá o RHS relevante ou usá-lo-á para passar a uma pergunta mais específica.

A ordem pela qual a colecção de regras está guardada é importante, tal como o é a forma como são tratadas as comparações LHS. Se existem várias condições LHS a cumprir antes de ser invocado um RHS, o sistema deve conhecer coisas como o número de comparações positivas necessárias, ou até que ponto se deve aproximar uma resposta de um dado resultado para constituir uma comparação positiva. Em alguns sistemas periciais, será lógico invocar um RHS assim que é encontrada uma correspondência. Noutros sistemas, onde uma única resposta do utilizador pode corresponder a vários LHS, estes podem encontrar-se organizados segundo uma ordem hierárquica, de tal modo que os de maior nível são invocados se possível antes de serem investigados os de nível inferior. O DENDRAL, um sistema baseado em regras que produz informações para químicos a partir de dados de espectrometria de massa, utiliza precisamente uma hierarquia de regras.

Em MECÂNICO, as regras encontram-se por uma ordem estrita, existindo um trajecto único para a regra seguinte. Este trajecto é de facto o RHS da regra.

Vejamos como tudo isto funciona. Começa-se por um menú de possibilidades:

```
TEM PORTANTO PROBLEMAS COMO SEU AUTOMÓVEL.
VEJAMOS SE É POSSÍVEL DESCOBRIR A CAUSA...
ESCREVA UMA LETRA QUE DESCREVA O TIPO DE PROBLEMA:
A - O MOTOR NÃO RODA
B - O MOTOR RODA, MAS NÃO ARRANCA
C - O MOTOR ARRANCA MAS PÁRA IMEDIATAMENTE A SEGUIR
D - O MOTOR FUNCIONA, MAS VAI ABAIXO MUITAS VEZES
E - O MOTOR FUNCIONA, MAS É INSTÁVEL EM PONTO MORTO
```

Digamos que o nosso problema é o segundo: O MOTOR RODA, MAS NÃO ARRANCA. Depois de indicarmos isto ao computador, este conduz-nos por uma série de questões, indicando depois as sugestões apropriadas:

```
                            > OK          S <

VERIFIQUE SE AS EXTREMIDADES DOS CABOS NO DISTRIBUIDOR
ESTÃO HUMIDAS.É POSSÍVEL EXISTIREM ÁREAS COM HUMIDADE?
                          (S - SIM, N - NÃO)?
                            > OK          S <

UTILIZE UM PULVERIZADOR CONTRA A HUMIDADE.

É VISÍVEL POEIRA NA PARTE ISOLANTE DO ENROLAMENTO
                  OU NA TAMPA DO DISTRIBUIDOR?
                          (S - SIM, N - NÃO)?
                            > OK          S <
```

```
LIMPE O INTERIOR E EXTERIOR DO DISTRIBUIDOR

VERIFIQUE SE TODOS OS CABOS ESTÃO FIXOS E SECOS ANTES DE CONTINUAR

SE O MOTOR NÃO ARRANCA, DEVE VERIFICAR AS VELAS:

A FAÍSCA NA EXTREMIDADE DA VELA SALTA MAIS DE 1 CM?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <
```

A ordem pela qual o sistema pericial nos apresenta as sugestões depende das respostas que lhe damos:

```
OBSERVA FAÍSCA NAS VELAS?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <

AS VELAS ESTÃO ENGORDURADAS?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <

SE SE TRATA DE UMA EMERGÊNCIA, EXPERIMENTE DIMINUIR A DISTÂNCIA
ENTRE OS PLATINADOS PARA METADE DO NORMAL

TEM GASOLINA NO DEPÓSITO?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <

OS TUBOS DE ALIMENTAÇÃO E VÁCUO ESTÃO BEM SEGUROS?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <

REMOVA O FILTRO DE AR DO CARBURADOR
PARECE-LHE SECO?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <

ACCIONOU O MOTOR DE ARRANQUE MUITAS VEZES NOS ÚLTIMOS MINUTOS?
                                   (S - SIM, N - NÃO)?
                                    > OK          S <

   ESPERE CERCA DE UM MINUTO, CARREGUE NO ACELERADOR E MANTENHA-O
NA MESMA POSIÇÃO, SEM BOMBEAR. DEVE CONSEGUIR ARRANCAR.
```

Se o leitor tem problemas ao tentar arrancar o motor do seu automóvel, e se gostaria que o seu computador o ajudasse a evitar custos de oficina, introduza na máquina a listagem seguinte e passará a dispor de um MECÂNICO privativo:

```
10 REM MECÂNICO
20 CLS
30 PRINT ''TEM PORTANTO PROBLEMAS COM O SEU AUTOMÓVEL.''
40 PRINT ''VEJAMOS SE É POSSÍVEL DESCOBRIR A CAUSA...''
50 PRINT
60 PRINT ''ESCREVA UMA LETRA QUE DESCREVA O TIPO DE PROBLEMA:''
70 PRINT
80 PRINT '' A - O MOTOR NÃO RODA''
90 PRINT '' B - O MOTOR RODA, MAS NÃO ARRANCA''
100 PRINT '' C - O MOTOR ARRANCA MAS PÁRA IMEDIATAMENTE A
    SEGUIR''
110 PRINT '' D - O MOTOR FUNCIONA, MAS VAI ABAIXO MUITAS VEZES''
120 PRINT '' E - O MOTOR FUNCIONA, MAS É INSTÁVEL EM PONTO
    MORTO''
130 IF INKEY$ < > '' '' THEN 130
140 A$=INKEY$
150 IF A$ < ''A''  OR A$ >  ''E'' THEN 140
160 GOSUB 1560
170 ON ASC(A$)-64 GOTO 200,580,1200,1230,1350
180 END
190 REM * * * * * * * * * * * * * *
200 REM NÃO RODA
210 PRINT ''COMEÇAREMOS POR VERIFICAR A BATERIA''
220 PRINT ''LIGUE AS LUZES''
230 PRINT TAB(6);''PARECEM-LHE FRACAS?''
240 GOSUB 1520
250 IF A$=''N'' THEN 350: REM BATERIA OK
260 PRINT ''OS CABOS DA BATERIA ESTÃO MAL FIXOS OU CORROÍDOS?''
270 GOSUB 1520
280 IF A$=''S'' THEN PRINT ''APERTE-OS E LIMPE-OS''
290 GOSUB 1580
300 PRINT ''A CORREIA ESTÁ MUITO FOLGADA?''
310 GOSUB 1520
320 IF A$=''S'' THEN PRINT TAB(8);''TIRE A FOLGA À CORREIA''
330 GOSUB 1580
340 PRINT ''CABOS LIGADOS A OUTRA BATERIA OU UM EMPURRÃO DEVEM
    ARRANCAR O MOTOR'': END
350 PRINT ''O CABO DA BATERIA APRESENTA-SE DESAPERTADO''
360 PRINT ''OU CORROÍDO?''
370 GOSUB 1520
380 IF A$=''S'' THEN PRINT ''APERTE E LIMPE AS LIGAÇÕES''
```

```
390 GOSUB 1580
400 PRINT ''COLOQUE UMA PEÇA METÁLICA ENTRE OS TERMINAIS''
410 PRINT ''DO SOLENÓIDE. O MOTOR DE ARRRANQUE FUNCIONA?''
420 GOSUB 1520
430 IF A$=''S'' THEN PRINT ''O INTERRUPTOR DE IGNIÇÃO ESTÁ
    PROVAVELMENTE DEFICIENTE''
440 GOSUB 1580
450 IF A$=''S'' THEN PRINT TAB(4);''DEVE SER SUBSTITUÍDO'': END
460 PRINT ''O MOTOR DE ARRANQUE REAGE?''
470 GOSUB 1520
480 IF A$=''S'' THEN PRINT ''O MOTOR DE ARRANQUE PODE ESTAR AVA-
    RIADO'': GOTO 520
490 PRINT ''COMO NADA ACONTECE, O SOLENÓIDE ESTÁ PROVAVELMENTE
    DEFICIENTE''
500 PRINT ''UM EMPURRÃO DEVE BASTAR PARA FAZER ARRANCAR''
510 PRINT ''O SEU AUTOMÓVEL.'': END
520 REM MOTOR DE ARRANQUE AVARIADO
530 PRINT ''DESLIGUE A IGNIÇÃO, E META UMA MUDANÇA ALTA''
540 PRINT ''EMPURRE O CARRO 30 CENTÍMETROS A FIM DE''
550 PRINT ''O LIBERTAR'': END
560 RETURN
570 REM * * * * * * * * * * * * * *
580 REM RODA, NÃO ARRANCA
590 PRINT ''VERIFIQUE SE AS EXTREMIDADES DOS CABOS NO
    DISTRIBUIDOR ESTÃO HÚMIDAS''
600 PRINT'' É POSSÍVEL EXISTIREM ÁREAS COM HUMIDADE?''
610 GOSUB 1520
620 IF A$=''S'' THEN PRINT ''UTILIZE UM PULVERIZADOR CONTRA
    A HUMIDADE'': GOSUB 1580
630 PRINT ''É VISÍVEL POEIRA NA PARTE ISOLANTE DO ENROLAMENTO
    OU NA TAMPA''
650 PRINT TAB(11);''DO DISTRIBUIDOR?''
660 GOSUB 1520
670 IF A$=''S'' THEN PRINT ''LIMPE O INTERIOR E O EXTERIOR''
680 IF A$=''S'' THEN PRINT ''DO DISTRIBUIDOR.'': GOSUB 1580
690 PRINT ''VERIFIQUE SE TODOS OS CABOS ESTÃO FIXOS
    E SECOS ANTES DE CONTINUAR''
700 GOSUB 1580
710 PRINT ''SE O MOTOR NÃO ARRANCA, DEVE VERIFICAR AS VELAS:''
720 GOSUB 1580
730 PRINT ''A FAÍSCA NA EXTREMIDADE DO CABO DA VELA SALTA
    MAIS DE 1 CM?''
740 GOSUB 1520
750 IF A$=''S'' THEN PRINT ''AS VELAS ESTÃO DEFICIENTES'':
    GOSUB 1580
```

```
760 IF A$=''N'' THEN 830
770 PRINT ''AS VELAS ESTÃO ENGORDURADAS?''
780 GOSUB 1520
790 IF A$=''S'' THEN PRINT ''NÃO SE PODE FAZER UMA REPARAÇÃO DE
    EMERGÊNCIA''
800 IF A$=''S'' THEN PRINT ''ENQUANTO AS VELAS ESTIVEREM NESSE
    ESTADO'': GOSUB 1580: GOTO 770
810 IF A$=''N'' THEN PRINT ''SE SE TRATA DE UMA EMERGÊNCIA,
    EXPERIMENTE DIMINUIR''
820 IF A$=''N'' THEN PRINT '' A DISTÂNCIA ENTRE OS PLATINADOS
    PARA METADE DO NORMAL'': GOSUB 1580: GOTO 910
830 PRINT ''OBSERVA FAÍSCA NAS VELAS?''
840 GOSUB 1520
850 IF A$=''S'' THEN 770
860 GOSUB 870: END
870 PRINT ''VERIFIQUE O ROTOR, A BOBINA E A TAMPA
    DO DISTRIBUIDOR''
880 PRINT ''OBSERVANDO SE APRESENTAM FENDAS. SE NÃO EXISTIREM,''
890 PRINT ''É PROVÁVEL QUE OS PLATINADOS OU CONDENSADOR ESTEJAM
    DEFICIENTES''
900 PRINT ''DEVE DIRIGIR-SE A UMA OFICINA'': RETURN
910 PRINT ''TEM GASOLINA NO DEPÓSITO?''
920 GOSUB 1520
930 IF A$=''N'' THEN PRINT ''ENCHA O DEPÓSITO E TENTE DE NOVO'':
    GOSUB 1580
940 PRINT ''OS TUBOS DE ALIMENTAÇÃO E VÁCUO ESTÃO BEM SEGUROS?''
950 GOSUB 1520
960 IF A$=''N'' THEN PRINT ''FIXE-AS BEM E TENTE DE NOVO'':
    GOSUB 1580
970 PRINT ''REMOVA O FILTRO DE AR DO CARBURADOR''
980 GOSUB 1580
990 PRINT ''PARECE-LHE SECO?''
1000 GOSUB 1520
1010 IF A$=''N'' THEN 1080
1020 PRINT ''RODE O MOTOR ALGUMAS VEZES TAPANDO COM A MÃO''
1030 PRINT ''A ENTRADA DE AR'': GOSUB 1580
1040 PRINT ''TEM A MÃO HÚMIDA DE GASOLINA?''
1050 GOSUB 1520
1060 IF A$=''N'' THEN PRINT ''DESAPERTE A TAMPA DO DEPÓSITO SE A
     VENTILAÇÃO ESTIVER ENTUPIDA''
1070 IF A$=''N'' THEN PRINT ''A BOMBA DE GASOLINA PODE NÃO ESTAR
     A FUNCIONAR'': GOSUB 900: END
1080 PRINT ''ACCIONOU O MOTOR DE ARRANQUE MUITAS VEZES''
1090 PRINT ''NOS ÚLTIMOS MINUTOS?''
1100 GOSUB 1520
```

```
1110 IF A$=''N'' THEN 1150
1120 PRINT ''ESPERE CERCA DE UM MINUTO, CARREGUE NO ACELERADOR
     E''
1130 PRINT ''MANTENHA-O NA MESMA POSIÇÃO, SEM O BOMBEAR. DEVE
     CONSEGUIR''
1140 PRINT ''ARRANCAR .'': END
1150 PRINT ''O MOTOR PODE ESTAR AFOGADO NESTE MOMENTO, E A
     BORBOLETA''
1160 PRINT ''PODE ESTAR PRESA.'': GOSUB 1580
1170 PRINT ''DÊ UMA PANCADA NO LADO DO CARBURADOR E TENTE
     NOVAMENTE''
1180 PRINT ''FAZER PEGAR O MOTOR'': END
1190 REM* * * * * *
1200 REM ARRANCA, MAS PÁRA
1210 PRINT ''ISTO SUGERE UMA RESISTÊNCIA DEFEITUOSA QUE DEVE SER
     SUBSTITUÍDA'': END
1120 REM * * * * * * * * * * * * * * *
1230 REM FUNCIONA, VAI ABAIXO
1240 PRINT ''O PROBLEMA PODE SER CAUSADO POR:''
1250 PRINT TAB(7);''CABOS EM CURTO-CIRCUITO (OU MAL FIXOS);''
1260 PRINT TAB(7);''UMA FALHA NO SISTEMA DE ALIMENTAÇÃO''
1280 PRINT ''PROCURE PRIMEIRO CABOS MAL FIXOS OU EM
     CURTO-CIRCUITO'': GOSUB 1580
1290 PRINT ''SE NÃO ESTIVEREM BEM, REPARE-OS. SE ESTIVEREM,
     TALVEZ O PROBLEMA ESTEJA NAS VELAS''
1300 GOSUB 870
1310 PRINT ''PODE AINDA FAZER UMA VERIFICAÇÃO FINAL DAS VELAS''
1320 GOSUB 1580
1330 GOTO 1360
1340 REM * * * * * * * * * * * * * * *
1350 REM FUNCIONA, IRREGULAR EM PONTO MORTO
1360 PRINT ''TALVEZ UMA DAS VELAS ESTEJA DEFEITUOSA''
1370 GOSUB 1580
1380 PRINT ''DESLIGUE-OS UM DE CADA VEZ. OS QUE''
1390 PRINT ''NÃO FIZEREM O MOTOR IR ABAIXO EM PONTO MORTO''
1400 PRINT ''ESTÃO DEFEITUOSOS. REPARE-OS E DEPOIS''
1410 PRINT ''CONTINUE'': GOSUB 1580
1420 PRINT ''DESCOBRIU ALGUMA VELA DEFEITUOSA?''
1430 GOSUB 1520
1440 IF A$=''S'' THEN PRINT ''SUBSTITUA TODAS AS VELAS SE FOR
     POSSÍVEL,''
1450 IF A$=''S'' THEN PRINT ''OU APENAS AS QUE DESCOBRIU ESTAREM
     DEFEITUOSAS'': GOSUB 1580
1460 PRINT ''A CAUSA MAIS COMUM DE UM MAU PONTO MORTO,''
1470 PRINT ''CONSIDERANDO QUE AS VELAS ESTÃO BEM,''
```

```
1480 PRINT ''É QUE A MISTURA É DEMASIADO RICA''
1490 PRINT ''DEVE PORTANTO CORRIGI-LA'': END
1500 PRINT ''AJUSTE A VELOCIDADE DE PONTO MORTO ACTUANDO SOBRE
     O JIGGLER'': END
1510 REM * * * * * * * * * * * * * *
1520 PRINT TAB(16);(S - SIM, N - NÃO)?''
1530 IF INKEY$ < > '' ''THEN 1530
1540 A$=INKEY$
1550 IF A$ < > ''S'' AND A$ < > ''N'' THEN 1540
1560 PRINT TAB(22);'' > OK   '';A$;'' < ''
1570 BEEP: REM UTILIZE UMA INSTRUÇÃO QUE PRODUZA UM SOM NESTA
     LINHA
1580 FOR T=1 TO 1000:NEXT T
1590 PRINT
1600 RETURN
```

Além de compreendermos como pode funcionar um sistema baseado em regras usando este programa, espero que o leitor se aperceba das desvantagens de um sistema cujas regras sejam definidas por ligações deterministas. São extremamente difíceis de modificar, sem destruir quase completamente o programa. Pode ser igualmente difícil descobrir um erro de julgamento do sistema, e este não poderá aprender por si mesmo como acontece noutros programas (como RITA, que apresentaremos mais adiante).

Apesar destas desvantagens, em muitas situações um sistema determinista como MECÂNICO pode ser bastante mais útil do que um sistema geral.

O leitor deve pesar estas vantagens e desvantagens quando determina o tipo de sistema pericial que poderá resolver o seu problema.

5

O DOUTOR JÁ CHEGOU

No meu livro «Explorando a Inteligência Artificial no seu Microcomputador» apresentei um sistema pericial fictício, chamado MEDICI, que a partir das respostas às suas perguntas dava sugestões relativas à saúde e longevidade. Decidi criar um novo MEDICI para este livro.

A melhor maneira de viver muito tempo consiste em escolher pais e avós que tenham vivido muito. Como isto não é possível, convirá recorrer à melhor solução seguinte, que consiste em aplicar no nosso quotidiano os conhecimentos já acumulados pelo homem no domínio da nutrição e da saúde. É certo que poucos vivem da forma mais inteligente, ou aplicam toda a informação que está ao nosso dispor. Quanto mais não seja, seria provavelmente aborrecido viver dessa maneira.

MEDICI poderá dizer-lhe como vive de facto, em vez de perguntar quais as regras que aplica ou não na sua vida. E depois de este sistema lhe ter aplicado uma «classificação» em termos de saúde e comentado o estado desta, indicar-lhe-á o número de anos que viverá se não reformar os seus hábitos. Pode então executar de novo o sistema, e ver como conseguirá alterar a sua expectativa de vida modificando os hábitos. Pode aproveitar a sua vida mais extensa para escrever mais sistemas periciais...

Experimentei MEDICI num voluntário à força e cheguei aos seguintes resultados:

```
QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS DA VERDADE
(ESCOLHA SÓ UMA):

A - SOU FRANCAMENTE GORDO
B - SOU GORDO
C - SOU UM POUCO FORTE
D - TENHO MAIS OU MENOS O PESO CERTO
E - SOU MAIS MAGRO DO QUE DEVERIA

                    OK     C

-5
```

QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS DA VERDADE (ESCOLHA SÓ UMA):
FAÇO EXERCÍCIO, QUE ELEVA A MINHA PULSAÇÃO PARA 120 OU MAIS, DURANTE PELO MENOS O SEGUINTE NÚMERO DE HORAS POR SEMANA:

A - MENOS DE UM QUARTO DE HORA
B - ENTRE UM QUARTO E TRÊS QUARTOS DE HORA
C - ENTRE TRÊS QUARTOS DE HORA E UMA HORA E MEIA
D - ENTRE UMA HORA E MEIA E DUAS HORAS E MEIA
E - MAIS DE DUAS HORAS E MEIA

OK A

0

QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS DA VERDADE (ESCOLHA SÓ UMA):

QUANDO GUIO:
A - QUASE NUNCA USO O CINTO DE SEGURANÇA
B - USO-O UM QUARTO DO TEMPO EM QUE ESTOU A CONDUZIR
C - USO O CINTO DE SEGURANÇA EM METADE DAS DESLOCAÇÕES QUE FAÇO
D - USO O CINTO DE SEGURANÇA EM QUASE TODAS AS MINHAS DESLOCAÇÕES
E - USO SEMPRE O CINTO DE SEGURANÇA

OK E

8

QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS DA VERDADE (ESCOLHA SÓ UMA):

RECORDO AS REGRAS DA NUTRIÇÃO E TENTO OBEDECER-LHES:

A - SEMPRE
B - QUASE SEMPRE
C - GERALMENTE
D - ÀS VEZES
E - QUASE NUNCA

OK D

1

QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS DA VERDADE (ESCOLHA SÓ UMA):

FUMA (UM CHARUTO CONTA COMO UM CIGARRO)?

A - NÃO
B - MENOS DE QUINZE CIGARROS POR DIA
C - 15 A 25 CIGARROS POR DIA
D - 26 A 42 CIGARROS POR DIA
E - MAIS DE 42 CIGARROS POR DIA

OK A

0

QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS DA VERDADE (ESCOLHA SÓ UMA):

BEBE? QUANTOS ''DRINKS'' (EM MÉDIA) POR DIA?

A - NENHUM
B - MENOS DE TRÊS
C - 3 A 6
D - 7 A 9
E - MAIS DE 9

OK C

-6

OPINIÃO PESSOAL DE MEDICI:

PESO:-5
EXERCÍCIO: 0
SEGURANÇA NA CONDUÇÃO: 8
NUTRIÇÃO: 1
FUMO: 0
ALCOOL:-6
STRESS:-8

A SUA CLASSIFICAÇÃO É -10

NUMA ESCALA EM QUE A MÉDIA CORRESPONDE A ZERO, O MENOR VALOR É CERCA DE - 80, E O MAIOR VALOR É CERCA DE 30

```
ISTO INDICA QUE O SEU ESTADO DE SAÚDE É ABAIXO DA MÉDIA

EXPECTATIVA DE VIDA:

  HOMEM           MULHER
  60 A 66         65 a 71
```

Vejamos agora a listagem, para que o leitor possa aplicar este teste a si mesmo:

```
10 REM MEDICI
20 CLS
30 GOSUB 1250
40 PRINT ''A - SOU FRANCAMENTE GORDO''
50 PRINT ''B - SOU GORDO''
60 PRINT ''C - SOU UM POUCO FORTE''
70 PRINT ''D - TENHO MAIS OU MENOS O PESO CERTO''
80 PRINT ''E - SOU MAIS MAGRO DO QUE DEVERIA''
90 GOSUB 1170
100 WEIGHT=5* (ASC(A$)-68):IF A$=''E'' THEN WEIGHT=0
110 PRINT WEIGHT
120 GOSUB 1250
140 PRINT ''FAÇO EXERCÍCIO, QUE ELEVA A MINHA PULSAÇÃO PARA
    120 OU MAIS,''
150 PRINT ''DURANTE PELO MENOS O SEGUINTE NÚMERO DE HORAS POR
    SEMANA:''
170 PRINT
180 PRINT ''A - MENOS DE UM QUARTO DE HORA''
190 PRINT ''B - ENTRE UM QUARTO E TRÊS QUARTOS DE HORA''
200 PRINT ''C - ENTRE TRÊS QUARTOS DE HORA E UMA HORA E MEIA''
210 PRINT ''D - ENTRE UMA HORA E MEIA E DUAS HORAS E MEIA''
220 PRINT ''E - MAIS DE DUAS HORAS E MEIA''
230 GOSUB 1170
240 EXERCISE=5* (ASC(A$)-63)-5:IF A$=''A'' THEN EXERCICE=0
250 PRINT EXERCISE
260 GOSUB 1250
270 PRINT ''QUANDO GUIO:'':PRINT
280 PRINT ''A - QUASE NUNCA USO O CINTO DE SEGURANÇA''
290 PRINT ''B - USO-O UM QUARTO DO TEMPO EM QUE ESTOU A
    CONDUZIR''
300 PRINT ''C - USO O CINTO DE SEGURANÇA EM METADE DAS
    DESLOCAÇÕES QUE FAÇO''
310 PRINT ''D - USO O CINTO DE SEGURANÇA EM QUASE TODAS AS
    MINHAS DESLOCAÇÕES''
320 PRINT ''E - USO SEMPRE O CINTO DE SEGURANÇA''
```

```
330 GOSUB 1170
340 SEATBELT=2*(ASC(A$)-65)
350 PRINT SEATBELT
360 GOSUB 1250
370 PRINT ''RECORDO AS REGRAS DA NUTRIÇÃO E TENTO OBEDECER-
    -LHES:''
380 PRINT
390 PRINT ''A - SEMPRE''
400 PRINT ''B - QUASE SEMPRE''
410 PRINT ''C - GERALMENTE''
420 PRINT ''D - ÀS VEZES''
430 PRINT ''E - QUASE NUNCA''
440 GOSUB 1170
450 DIET=-ASC(A$)+69
460 PRINT DIET
470 GOSUB 1250
480 PRINT ''FUMA (UM CHARUTO CONTA COMO UM CIGARRO)?''
490 PRINT
500 PRINT ''A - NÃO''
510 PRINT ''B - MENOS DE QUINZE CIGARROS POR DIA''
520 PRINT ''C - 15 A 25 CIGARROS POR DIA''
530 PRINT ''D - 26 A 42 CIGARROS POR DIA''
540 PRINT ''E - MAIS DE 42 CIGARROS POR DIA''
550 GOSUB 1170
560 SMOKING=-7*(ASC(A$)-65)
570 PRINT SMOKING
580 GOSUB 1250
590 PRINT ''BEBE? QUANTOS 'DRINKS' (EM MÉDIA) POR DIA?''
600 PRINT
610 PRINT ''A - NENHUM''
620 PRINT ''B - MENOS DE TRÊS''
630 PRINT ''C - 3 A 6''
640 PRINT ''D - 7 A 9''
650 PRINT ''E - MAIS DE 9''
660 GOSUB 1170
670 DRINK= -30
680 IF A$=''A'' THEN DRINK=0
690 IF A$=''B'' THEN DRINK=1
700 IF A$=''C'' THEN DRINK=DRINK/5
710 IF A$=''D'' THEN DRINK=DRINK/2
720 PRINT DRINK
730 GOSUB 1250
740 PRINT ''EM GERAL, PENSA TER SOFRIDO BASTANTE DE STRESS'':
750 PRINT ''NOS ÚLTIMOS SEIS MESES?''
760 PRINT
```

```
770 PRINT ''A - MUITO STRESS''
780 PRINT ''B - ALGUM STRESS''
790 PRINT ''C - POUCO STRESS''
800 PRINT ''D - NEUTRO''
810 PRINT ''E - SEM STRESS''
820 GOSUB 1170
830 STRESS=INT (2.5 * (ASC(A$) - 69))
840 PRINT STRESS
850 GOSUB 1300: CLS
860 PRINT ''OPINIÃO PESSOAL DE MEDICI:''
870 PRINT
880 PRINT ''PESO: '';WEIGHT
890 PRINT ''EXERCÍCIO: '';EXERCISE
900 PRINT ''SEGURANÇA NA CONDUÇÃO: '';SEATBELT
910 PRINT ''NUTRIÇÃO: '';DIET
920 PRINT ''FUMO: '';SMOKING
930 PRINT ''ALCOOL: '';DRINK
940 PRINT ''STRESS: '';STRESS
950 GOSUB 1300
960 ANT=WEIGHT+EXERCISE+SEATBELT+DIET+SMOKING+DRINK+STRESS
970 GOSUB 1300: PRINT
980 PRINT ''A SUA CLASSIFICAÇÃO É ''ANT: PRINT
990 PRINT ''NUMA ESCALA EM QUE A MÉDIA CORRESPONDE A ZERO; O
    MENOR VALOR''
1000 PRINT ''É CERCA DE -80, E O MAIOR VALOR É CERCA DE 30''
1010 GOSUB 1300
1020 PRINT
1030 IF ANT<6 AND ANT>-6 THEN A$=''MÉDIO'': L$=''62 A 73
     72 A 78''
1040 IF ANT <-5 AND ANT>-21 THEN A$=''ABAIXO DA MÉDIA'':
     L$=''60 A 66    65 A 71''
1050 IF ANT<-20 THEN A$=''FRACO'':L$=''60 OU MENOS
     65 OU MENOS''
1060 IF ANT<-45 THEN A$=''MUITO FRACO''
1070 IF ANT<-60 THEN A$=''MUITÍSSIMO FRACO''
1080 IF ANT>5 AND ANT<15 THEN A$=''BOM'':L$=''74 A 80
     79 A 85''
1090 IF ANT>14 THEN A$=''MUITO BOM'':L$=''81 OU MAIS
     86 OU MAIS''
1100 PRINT ''ISTO INDICA QUE O SEU ESTADO DE SAÚDE É '';A$
1110 PRINT
1120 PRINT ''EXPECTATIVA DE VIDA:''
1130 PRINT TAB(3);''HOMEM    MULHER''
1140 PRINT TAB(3);L$
1150 END
```

```
1160 REM* * * * * * * * * * * * * *
1170 REM ACEITAR ENTRADAS
1180 IF INKEY$< >'' ''THEN 1180
1190 A$=INKEY$
1200 IF A$<''A'' OR A$> ''E'' THEN 1190
1210 PRINT:PRINT TAB(12);''OK     '';A$
1220 RETURN
1230 REM
1240 REM ATRASO
1250 FOR J=1 TO 1000: NEXT J
1260 CLS
1270 PRINT:PRINT:PRINT:PRINT
1280 PRINT ''QUAL DAS SEGUINTES AFIRMAÇÕES SE APROXIMA MAIS
     DA VERDADE''
1290 PRINT ''(ESCOLHA SÓ UMA):''
1300 PRINT
1310 FOR J=1 TO 400: NEXT J
1320 RETURN
```

6
DESENVOLVENDO UM SISTEMA PERICIAL

Se bem que programas como MECÂNICO e MEDICI tenham interesse, e demonstrem claramente como funcionam muitos dos sistemas periciais existentes no Mundo, apresentam duas desvantagens principais. A primeira é que só codificam conhecimentos sobre um tema específico; estão restringidos a um domínio. A segunda desvantagem é que antes de o programa ser escrito, alguém deve ter os conhecimentos e saber as respostas para que a base de conhecimentos da máquina possa ser formada.

Se o leitor já sabe como deve enfrentar os pequenos problemas do seu automóvel, é provável que não lhe interesse ligar o computador apenas para ficar a saber o que já sabe.

No entanto, como já disse anteriormente, muitos sistemas periciais só foram desenvolvidos depois de os conhecimentos de um ser humano terem sido codificados e organizados por um «engenheiro de conhecimento». Este tenta determinar a heurística através da qual o problema é resolvido por um ser humano (heurística designa aqui o trajecto com vista a um fim que foi descoberto através da experiência e não pelo cálculo. Um trajecto deste tipo não produz necessariamente resultados, se bem que a experiência possa ter demonstrado que na maioria dos casos se aproxima pelo menos do que se pretende. Os programas de xadrez funcionam, em grande medida, de modo heurístico. O método algorítmico, ao contrário do heurístico, é aquele em que se aplica uma técnica que produz inevitavelmente um dado resultado. O seu computador utiliza algoritmos nele incorporados para, por exemplo, somar números).

Os conhecimentos organizados de um especialista humano são preservados no interior do programa; pensemos no entanto no interesse que haveria em criar um programa que fosse capaz de deduzir por si mesmo o conjunto de regras mais apropriadas, partindo de dados grosseiros. Seria ainda mais interessante criar um programa capaz de actuar deste modo num campo em que o próprio especialista humano não dispõe de regras suficientes.

Se fosse possível fazer isto — criar um programa capaz de produzir regras próprias a partir de dados grosseiros, e usá-lo para atingir

resultados satisfatórios onde os humanos não os conseguissem — seria extremamente importante que o programa pudesse depois explicar a natureza das regras por ele desenvolvidas, a fim de permitir aos seres humanos aplicá-las. O sistema pericial transformar-se-ia assim numa espécie de assistente de investigação e num instrutor.

Vamos escrever um programa deste tipo neste capítulo. Se bem que os especialistas em estatísticas possam ficar um tanto horrorizados com a forma pouco rigorosa como os números são manipulados pelo programa, o valor de um sistema pericial mede-se pelos seus resultados. Se pode especializar-se em quase tudo, incluindo campos onde os seres humanos não conseguem criar heurísticas fiáveis, então pouco interessa aquilo que faz aos números.

Conheça RITA

O nosso sistema tem ainda uma outra característica. Muitas decisões enfrentadas pelos seres humanos não permitem resultados completamente claros. Se bem que a resposta à pergunta «O número A é maior do que o número B?» só possa ter uma resposta, uma pergunta do tipo «A vitamina X é boa para si?» desloca-nos rapidamente para o mundo do «depende». A vitamina X é muitas vezes boa para recém-nascidos, excepto quando possuem sangue do tipo ES; se as mães praticam karate, este facto pode modificar o valor da vitamina para os seus bebés. Esta regra «prática» inclui «muitas vezes», «excepto quanto» e «pode modificar»; todos eles conceitos bastante «difusos».

RITA está à vontade num mundo em que a causa e o efeito nem sempre podem ser colocadas em categorias óbvias. Note-se que o nome de Rita se deve a uma comédia que teve muito êxito em Londres. «A Educação de Rita». O nosso programa pode ser educado, tal como indica a sub-rotina que se inicia na linha 970. A propósito deste programa empregaremos muitas vezes o termo «difusão», dada a indefinição das regras usadas. O programa dará porém resultados aceitáveis mesmo num mundo de respostas sim/não.

Os componentes de um sistema pericial

Antes de continuarmos a desenvolver o nosso sistema, voltaremos ao que já foi dito sobre as partes principais dos sistemas periciais em geral. A maior parte dos disponíveis comercialmente possuam duas componentes principais, o motor de inferência e a base de conhecimentos. O motor de inferência é o mecanismo através do qual são atingidas as conclusões. A base de conhecimento é o conjunto de dados grosseiros que o sistema manipula para tomar decisões.

Há grande interesse em manter separadas estas duas partes do sistema. Se o mecanismo de inferência pode ser usado em geral, poderá tratar diferentes bases de conhecimento em vários domínios, «pensando» em qualquer campo em função dos conhecimentos que são colocados à sua disposição.

Vimos já como isto poderia ser útil, quando discutimos o MYCIN, o potente sistema pericial desenvolvido em Stanford para diagnóstico de doenças infecciosas. Ao ser-lhe retirada a base de conhecimentos transformou-se no motor de inferência EMYCIN. A adição de novas bases de conhecimentos transformou o médico primeiro num mecânico de automóveis (com o auxílio do manual Pontiac), num consultor de análise estrutural (SACON), num outro médico especialista em hemorragias (CLOT), e num perito em campos intrigantes como os cobertos pelas designações LITHO, HEADMED e BLUEBOX, correspondentes a outros tantos sistemas.

RITA não pertence à mesma família de MYCIN e dos que nele se inspiram. Em vez de ter de ser alimentado em regras, RITA desenvolve regras próprias, modificando-as quando necessário se as conclusões a que chega estiverem erradas. Como é fascinante observar o desenvolvimento da sua base de regras, RITA permite ao utilizador observar o estado actual das regras depois de cada decisão ter sido tomada.

Um sistema geral

RITA foi criado especificamente como sistema «geral», capaz de tratar dados relativos quase a qualquer problema. Por outro lado, pode por vezes ser um pouco demorado a pedir informações antes de chegar a uma conclusão (se bem que seja capaz, quando são possíveis mais de duas conclusões, de determinar que certas perguntas não são relevantes para o resultado; nestes casos limitar-se-á a ignorá-las).

Portanto, se o leitor quiser usar RITA num campo específico, durante um extenso período de tempo, vale a pena gastar algum tempo no seu desenvolvimento a fim de obter os melhores resultados possíveis. Todos os sistemas periciais são testados entregando-lhes o estudo de exemplos já conhecidos, e comparando a resposta do sistema com os dados conhecidos. As respostas erradas sugerem a necessidade de alterar a base de regras e o leitor não deverá ter medo de fazer tais modificações.

Apesar disto, o leitor verificará que RITA funciona surpreendentemente bem na maior parte das situações. Tentei definir os limites que determinam o modo como RITA decide as suas conclusões «mais prováveis» ou as segundas mais prováveis, sujeitando o sistema a uma vasta gama de exemplos e alterando os valores em cada experiência.

Será um gato ou um cão?

Esta é um das perguntas que mais têm preocupado a humanidade desde o início dos tempos... Agora, com o auxílio de RITA, pode-se resolver o assunto de uma vez por todas.

Seguindo a actuação de RITA neste complexo exemplo deve ser fácil compreender como este programa funciona, e como pode ser usado para resolver os problemas que mais interessem ao leitor. O problema CÃO/GATO conduz a uma resposta «discreta»: ou é um cão ou um gato. Os diagnósticos médicos podem muitas vezes ser de outro tipo: pode tratar-se de uma infecção no sistema respiratório superior, ou uma forma suave de sarampo. RITA trata resultados de ambos os tipos, e não necessita de saber qual o campo a que o problema actual respeita. As respostas são sempre dadas na forma «O RESULTADO MAIS PROVÁVEL É...». Se os dados que foram comunicados a RITA, juntamente com a base de regras de que dispõe, sugerem que existem dois resultados igualmente prováveis, pode dar duas conclusões prefaciadas pelas palavras "O MAIS PROVÁVEL A SEGUIR É...». Depois de RITA ter aprendido a diferença entre um gato e um cão, obtemos geralmente uma única conclusão.

O programa começa por perguntar se queremos observar a base de conhecimentos actualizada após cada execução. Carregamos em qualquer tecla, antes de retorno:

```
CARREGUE EM QUALQUER TECLA, DEPOIS EM ''RETORNO''
SE QUISER OBSERVAR A BASE DE CONHECIMENTOS APÓS
CADA EXECUÇÃO; SENÃO CARREGUE APENAS EM ''RETORNO''

?L
```

Em seguida indicamos as «opções de saída» (isto é, as conclusões que o sistema pode atingir), teminando pela tecla de retorno. RITA, tal como se encontra, pode tratar até cinquenta resultados diferentes:

```
-----------------------------------------

INDIQUE OPÇÃO DE SAÍDA NÚMERO 1
    (CARREGUE <RETORNO> PARA TERMINAR)
? CÃO
-----------------------------------------

INDIQUE OPÇÃO DE SAÍDA NÚMERO 2
    (CARREGUE <RETORNO> PARA TERMINAR)
? GATO
-----------------------------------------
```

Depois de terminar a entrada de opções, RITA imprime-as no visor, e pede que introduzamos as «perguntas» que o sistema fará mais tarde.

RITA pode tratar até cinquenta perguntas:

```
? CÃO
? GATO
------------------------------------------------

INDIQUE PERGUNTA NÚMERO 1
    (CARREGUE EM RETORNO PARA TERMINAR)
?COME RATOS E FAZ MIAU
------------------------------------------------

INDIQUE PERGUNTA NÚMERO 2
    (CARREGUE EM RETORNO PARA TERMINAR)
?ANIMAL
------------------------------------------------

INDIQUE PERGUNTA NÚMERO 3
    (CARREGUE EM RETORNO PARA TERMINAR)
?LADRA CONTRA OS LADRÕES
------------------------------------------------
```

O leitor deve uma vez mais carregar em RETORNO para indicar que terminou a entrada de perguntas. Depois de o ter feito, RITA dir-lhe-á quais os temas sobre os quais poderá tomar decisões:

```
EIS OS OBJECTOS ENTRE OS QUAIS
POSSO DISCRIMINAR:

> CÃO
> GATO
------------------------------------------------

PENSE NUM DELES, E CARREGUE EM RETORNO
```

As perguntas são então apresentadas uma a uma, sendo pedido ao utilizador que responda a elas com um número entre 1 (significando 100% verdadeiro) e 0. Isto permite a RITA tratar dados directos (do tipo CÃO/GATO, PRETO/BRANCO) e dados que cobrem uma gama de valores (como peso, pressão sanguínea, etc).

No caso de dados que podem cobrir uma vasta gama de valores (como acontece com o peso), é fácil conceber um sistema que nos permite representar a menor leitura de peso como um zero, e a maior como um. O leitor observará, num exemplo dado adiante, que divido as temperaturas máxima e mínima por 10, significando que a grande maioria das temperaturas são indicadas por um número como .6 (indicando 6 graus). Qualquer resultado acima de 1 é indicado por 1. As horas de

luz solar são tratadas do mesmo modo, sendo 7 horas indicadas por .7, e assim por diante. Não interessa verdadeiramente quais são os números, desde que se invente uma regra que permita reduzi-los a uma gama entre 1 e 0 e se cumpra essa regra quaisquer que sejam os dados. Podemos adoptar regras diferentes para diversas categorias de dados num mesmo programa; RITA não se importa.

Podemos também usar o número introduzido para representar conceitos como «muito», «sempre» e «não importante» em resposta a perguntas como A FACE DO DOENTE ESTÁ CORADA (onde .9 pode representar «muito», e .1 pode representar «nada»); COSTUMA TER PROBLEMAS DE ARRANQUE DO AUTOMÓVEL (onde .9 indicará «sempre» e .4 «de vez em quando»), etc.

Os números introduzidos não têm de ser exactos. De facto, é difícil ver como se pode obter uma classificação coerente de um conceito como «quase nunca». No caso de RITA, uma indicação aproximada é quase sempre suficientemente boa, desde que a classificação seja coerente para uma dada categoria de dados.

Como se pode ver, isto significa que RITA é capaz de tratar praticamente qualquer problema que lhe apresentemos, quer os dados sejam do tipo SIM/NÃO, ou se distribuam por uma escala contínua (como a temperatura) ou sejam do tipo «significado variável». O nosso exemplo GATO/CÃO é, pelo menos para efeitos deste exemplo, do tipo SIM/NÃO.

Primeiro ensinamos algumas coisas à RITA sobre cães:

```
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

COME RATOS E FAZ MIAU? 0

-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

ANIMAL? 1
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

LADRA CONTRA OS LADRÕES? 1
-------------------------------------------

O RESULTADO MAIS PROVÁVEL É CÃO
O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
          (S OU N)? S
```

Por sorte, o sistema respondeu CÃO, a resposta correcta (de facto não foi sorte; como RITA não possui quaisquer informações em que possa fundamentar a decisão, limita-se a escolher a primeira opção).

RITA actualiza a base de regras, fazendo depois um relatório das suas descobertas:

```
-------------------------------------------

CÃO

COME RATOS E FAZ MIAU 0
ANIMAL 2
LADRA CONTRA OS LADRÕES 4

-------------------------------------------

GATO

COME RATOS E FAZ MIAU 0
ANIMAL 0
LADRA CONTRA OS LADRÕES 0

CARREGUE EM RETORNO PARA CONTINUAR O ENSINO
OU QUALQUER TECLA SEGUIDA DE RETORNO PARA USAR RITA
```

O programa atribuiu o valor 0 à pergunta COME RATOS E FAZ MIAU, 2 a ANIMAL e 4 a LADRA CONTRA OS LADRÕES. Isto não significa que pensa que a última questão seja mais importante do que ser ANIMAL, dado que não possui qualquer informação sobre a qual possa fundamentar as suas decisões. RITA multiplica o valor que atribui a cada resposta por um número que duplica para cada uma, pelo que o valor da primeira resposta é multiplicado por 1, o da segunda por 2, o da terceira por quatro, o da quarta por 8, etc. O leitor verifica que, até agora, a base de regras GATO está ainda cheia de zeros. Isto deve-se ao facto de RITA ainda não ter encontrado nenhum gato, nada sabendo sobre estes.

Apresentemos ao programa um gato, e vejamos o que ele responde:

```
-------------------------------------------

PENSE NUM DELES, E CARREGUE EM RETORNO
?
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

COME RATOS E FAZ MIAU? 1
```

```
---------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

ANIMAL? 1
---------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

LADRA CONTRA OS LADRÕES? 0
---------------------------------------------

O RESULTADO MAIS PROVÁVEL É CÃO

O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
            (S OU N)? N
---------------------------------------------
```

O sistema responde CÃO, dado que é a melhor comparação de que dispõe (a pergunta ANIMAL já respondida na execução anterior). Quando indicamos que CÃO está errado, RITA verifica quais são as outras opções. Se só existem duas, sabe que o resultado não apresentado é o correcto para os dados apresentados, e ajusta a regra da forma mais apropriada:

```
---------------------------------------------

CÃO

COME RATOS E FAZ MIAU 0
ANIMAL 2
LADRA CONTRA OS LADRÕES 4
---------------------------------------------

GATO

COME RATOS E FAZ MIAU 1
ANIMAL 2
LADRA CONTRA OS LADRÕES 0

CARREGUE EM RETORNO PARA CONTINUAR O ENSINO
OU QUALQUER TECLA SEGUIDA DE RETORNO PARA USAR RITA
```

Vemos que RITA criou a regra de que uma resposta positiva a COME RATOS E FAZ MIAU, e ANIMAL, produz gato, enquanto que as respostas positivas a ANIMAL e LADRA CONTRA OS CÃES produz cão. Daqui em diante, neste caso extremamente simples (dois resultados para perguntas SIM/NÃO) RITA deixará de cometer erros. Foi ensinada a distinguir dois animais.

Maior dificuldade

A situação já não é tão simples quando há mais de dois resultados possíveis, e pode ser necessário a RITA passar por mais sessões de treino para aprender o que deve. Mesmo então perguntará se a sua resposta está correcta (e se respondemos que não, perguntará qual é a mais provável a seguir), e modificará as suas regras ligeiramente no caso de ter cometido um erro.

No entanto, quando há mais de dois resultados, RITA aproveita uma sua outra habilidade. Pode decidir que certas perguntas são irrelevantes e que não ajudam a tirar conclusões (no exemplo CÃO/GATO, a pergunta sobre se é ou não ANIMAL não tem qualquer importância; não deu a RITA qualquer informação adicional que lhe permitisse decidir qual a criatura considerada).

Recorrendo a um texto elementar de química, decidi ensinar RITA a distinguir entre o magnésio, o ferro e o chumbo. Isto não é de facto muito difícil mesmo para ignorantes nestes assuntos, mas já é bastante diferente de distinguir entre um cão e um gato, um tema no qual me considero 100% especialista.

Começamos por indicar a RITA as opções de saída:

```
---------------------------------------------

INDIQUE OPÇÃO DE SAÍDA NÚMERO 1
    (CARREGUE EM RETORNO PARA TERMINAR)
? MAGNÉSIO
---------------------------------------------

INDIQUE OPÇÃO DE SAÍDA NÚMERO 2
    (CARREGUE EM RETORNO PARA TERMINAR)
? FERRO
---------------------------------------------

INDIQUE OPÇÃO DE SAÍDA NÚMERO 3
    (CARREGUE EM RETORNO PARA TERMINAR)
? CHUMBO
---------------------------------------------
```

Depois indicamos as perguntas que permitirão ao programa discriminar entre esses resultados:

```
---------------------------------------------

    INDIQUE PERGUNTA NÚMERO 1
    (CARREGUE EM RETORNO PARA TERMINAR)
? TEM UMA DENSIDADE INFERIOR A 8 G/CM3
```

```
-------------------------------------------
     INDIQUE PERGUNTA NÚMERO 2
     (CARREGUE EM RETORNO PARA TERMINAR)
? É UM METAL
-------------------------------------------
     INDIQUE PERGUNTA NÚMERO 3
     (CARREGUE EM RETORNO PARA TERMINAR)
? É VENENOSO
-------------------------------------------
     INDIQUE PERGUNTA NÚMERO 4
     (CARREGUE EM RETORNO PARA TERMINAR)
? CONDUZ ELECTRICIDADE
-------------------------------------------
     INDIQUE PERGUNTA NÚMERO 5
     (CARREGUE EM RETORNO PARA TERMINAR)
? BRILHA QUANDO POLIDO
-------------------------------------------
     INDIQUE PERGUNTA NÚMERO 6
     (CARREGUE EM RETORNO PARA TERMINAR)
? EMBACIA FACILMENTE
-------------------------------------------
```

RITA apresenta um relatório sobre a situação inicial, iniciando-se o treino do programa:

```
EIS OS OBJECTOS ENTRE OS QUAIS
POSSO DISCRIMINAR

> MAGNÉSIO
> FERRO
> CHUMBO


-------------------------------------------
PENSE NUM DELES E CARREGUE EM RETORNO
?
-------------------------------------------
 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

TEM DENSIDADE INFERIOR A 8 G/CM3? 0
-------------------------------------------
 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)
```

```
É UM METAL? 1
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

É VENENOSO? 1
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

CONDUZ ELECTRICIDADE? 1
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

BRILHA QUANDO POLIDO? 0
-------------------------------------------

 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

EMBACIA FACILMENTE? 1
-------------------------------------------

O RESULTADO MAIS PROVÁVEL É MAGNÉSIO
O SEGUINTE MAIS PROVÁVEL É CHUMBO

O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
            (S ou N)? N
O SEGUNDO RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
            (S OU N)? S
```

Repetimos as perguntas algumas vezes, até a base de conhecimento ter a seguinte aparência:

```
-------------------------------------------

MAGNÉSIO
TEM DENSIDADE INFERIOR A 8 G/CM3 1
É UM METAL 2
É VENENOSO 0
CONDUZ ELECTRICIDADE 8
BRILHA QUANDO POLIDO 0
EMBACIA FACILMENTE 32
```

```
--------------------------------------------
FERRO
TEM DENSIDADE INFERIOR A 8 G/CM3 1
É UM METAL 2
É VENENOSO 0
CONDUZ ELECTRICIDADE 8
BRILHA QUANDO POLIDO 16
EMBACIA FACILMENTE 32
--------------------------------------------
CHUMBO
TEM DENSIDADE INFERIOR A 8 G/CM3 0
É UM METAL 2
É VENENOSO 0
CONDUZ ELECTRICIDADE 8
BRILHA QUANDO POLIDO 0
EMBACIA FACILMENTE 32
```

Desta vez, antes de carregarmos em retorno para nova execução, carregamos noutra tecla a fim de indicar que o treino terminou e que RITA deve iniciar o seu trabalho:

```
CARREGUE EM RETORNO PARA CONTINUAR O TREINO
OU QUALQUER TECLA E RETORNO PARA USAR RITA
```

Um resultado prático da mudança de treino para trabalho, além de uma pequena alteração da «conversa», é que RITA decide agora quais as perguntas a fazer, em vez de as repetir todas:

```
EIS OS OBJECTOS ENTRE OS QUAIS POSSO DISCRIMINAR

> MAGNÉSIO
> FERRO
> CHUMBO
--------------------------------------------
ESTOU PRONTA A DETERMINAR QUAL O QUE VOCÊ TEM
 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

TEM DENSIDADE INFERIOR A 8 G/CM3? 1
ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

É VENENOSO? 0
 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)
```

```
BRILHA QUANDO POLIDO? 1

O RESULTADO MAIS PROVÁVEL É FERRO
O SEGUINTE MAIS PROVÁVEL É MAGNÉSIO

O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
           (S OU N)? S
```

Daqui em diante, a base de conhecimentos só será actualizada se for dito a RITA que as suas conclusões estão erradas.

```
TEM DENSIDADE INFERIOR A 8 G/CM3? 0
 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

É VENENOSO? 1
 ESCREVA UM NÚMERO ENTRE
1 (VERDADEIRO) E 0 (FALSO)   ($ PARA TERMINAR EXECUÇÃO)

BRILHA QUANDO POLIDO? 0

O RESULTADO MAIS PROVÁVEL É CHUMBO
O SEGUINTE MAIS PROVÁVEL É MAGNÉSIO

O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
         (S OU N)? S
```

7

RACIOCÍNIO «DIFUSO»

Se bem que seja razoavelmente fácil codificar certezas (SE ISTO é verdadeiro ENTÃO ISTO É SEMPRE verdadeiro) num programa de computador, a expressão de *graus* de certeza já não é tão simples. A lógica «difusa», um termo introduzido por L.A. Zadeh (1979), trata da tomada de decisões partindo de premissas a que falta rigor.

As conclusões tiradas são expressas em termos de *possibilidades* em vez de *probabilidades*, como as possíveis quando o grau de certeza de uma premissa pode ser expresso de modo rigoroso usando uma forma matemática. A maior parte dos mecanismos de raciocínio (como SE A É SEMPRE B, E C É A, ENTÃO C É B) são incapazes de tratar a falta de rigor e a simples possibilidade.

A falta de rigor está na base da maioria das acções humanas, desde a compreensão do discurso até à decisão sobre qual a jogada a fazer numa partida de xadrez. Os computadores não se dão bem, como acabamos de ver, com a falta de rigor. Um bit é 0 ou 1 (e nunca «possivelmente» 1). Os mecanismos de codificação que tratam as situações «difusas» exigem capacidades especiais dos programadores.

Em linguagens como a PROLOG e a HASTE (ver os capítulos 11 a 13) o grau de certezas é codificado directamente. Quando se usam linguagens declarativas deste tipo podemos recorrer a termos como «na maior parte dos casos«, «muitas vezes«, «normalmente», «por vezes», «quase nunca», etc. Mais ainda, os termos podem ser ligados e comparados entre si, de tal modo que podemos dizer por exemplo «Se é muito provável que A esteja presente, então B não estará geralmente presente».

Nas linguagens naturais, humanas, há um conjunto de termos de uso quotidiano que indica os graus de possibilidade correspondentes aos termos descritivos. O que se segue deve esclarecer o que pretendo dizer com isto. Pensemos num adjectivo, como «vermelho», «comprido», etc. Quando descrevemos um objecto como sendo vermelho ou comprido, precedemos muitas vezes tal descrição por uma palavra ou frase que modificam o adjectivo de algum modo, dando-lhe um sentido mais exacto. Entre estas palavras ou frases encontram-se «muito», «um pouco», «não particularmente», «de modo algum», e servem para atribuir ao adjectivo um determinado grau de probabilidade. Nestes ca-

sos, podemos afirmar que a «variável linguística» é formada por duas partes, um *termo primário* (por exemplo «vermelho») e um *modificador.*

Se bem que seja obviamente impossível apresentar um índice de probabilidade para cada uso de modificadores específicos (por exemplo, .95 para «muito» e .2 para «não particularmente») é possível na prática atribuir valores baseados na nossa compreensão do grau de «força» que um dado modificador dá a um adjectivo, e — o que é mais importante ainda — no nosso conhecimento do objecto que está a ser descrito e do significado do adjectivo.

Voltando ao mundo real

Os exemplos GATO/CÃO e MAGNÉSIO/FERRO/CHUMBO utilizados com o programa RITA eram bastante simples e exigiam apenas respostas do tipo SIM/NÃO. É chegado o momento de vermos se o programa pode funcionar igualmente bem usando dados menos definidos, e numa situação que não é fácil (ou é mesmo impossível) de determinar em termos de regras absolutas. A previsão do tempo é uma situação deste tipo. Se bem que possamos observar o céu de manhã e dizer qualquer coisa do tipo «Parece que vai chover» ou «Esperamos que não chova», dispomos provavelmente de poucos dados em que possamos basear uma previsão do resto do dia, ou do tempo que fará amanhã.

Vamos observar o tipo de clima em duas cidades diferentes, Londres e Melbourne. Veremos primeiro o tipo de variação ocorrida em Londres, usando quatro variáveis (horas de Sol, temperaturas máxima e mínima e milímetros de chuva). Apesar de só usarmos estas variáveis, o leitor verificará que RITA consegue dar resultados bastante bons. Examinando a sua «performance» neste campo poderemos compreender mais facilmente a forma como o programa trabalha.

Depois de observarmos o exemplo londrino, e analisarmos os passos do programa, veremos um conjunto de factos climáticos mais rigorosos relativamente a Melbourne, e notaremos como a qualidade dos resultados do programa aumenta quando este tem à sua disposição dados mais exactos.

Londres

Disse a RITA que pretendia escolher uma de três previsões para o tempo que faz num dado dia, a partir de informações relativas ao dia anterior. O programa deveria prever uma das seguintes situações:

— Chuva inferior a 1 mm
— Chuva superior a 4 mm
— Chuva entre 1 e 4 mm

As perguntas de discriminação seriam as seguintes:

— Temperatura mínima (dividida por 10, de modo a transformá-la num valor entre 0 e 1, sendo qualquer resultado superior a 1 indicado como 1).
— Temperatura máxima (valor manipulado do mesmo modo).
— Horas de Sol (divididas por 10; este processo não cria qualquer resposta superior a 1 — todos os valores neste exercício foram arredondados para a primeira casa decimal).

Depois de uma semana de entradas, tendo RITA encontrado dias correspondendo aos três resultados possíveis, a base de regras passou a ser a seguinte:

```
----------------------------------------------

CHUVA INFERIOR A 1 MM

TEMPERATURA MÍNIMA (C /10) .4
TEMPERATURA MÁXIMA (C /10) 1.9
QUEDA DE CHUVA (MM /10) .5
HORAS DE SOL (HORAS /10) 3.8

----------------------------------------------

CHUVA ACIMA DE 4 MM

TEMPERATURA MÍNIMA (C /10) .3
TEMPERATURA MÁXIMA (C /10) 1.8
QUEDA DE CHUVA (MM /10) 1.4
HORAS DE SOL (HORAS /10) 35.4

----------------------------------------------

CHUVA ENTRE 1 E 4 MM

TEMPERATURA MÍNIMA (C /10) .8
TEMPERATURA MÁXIMA (C/ 10) 2
QUEDA DE CHUVA (MM /10) 3.2
HORAS DE SOL (HORAS /10) .8

CARREGUE EM RETORNO PARA CONTINUAR O TREINO
OU QUALQUER OUTRA TECLA E RETORNO PARA USAR RITA
```

É bastante interessante examinar esta regra e observar as regras que RITA descobriu, vendo o modo como se comparam com as regras práticas que nós própios concebemos para analisar o tempo.

Para prever menos de 1 mm de chuva, o programa procura um valor de .4 para a temperatura mínima (e quase o mesmo, .3, quando procura mais de 4 mm de chuva) enquanto aceita .8 como um valor que tende a fundamentar uma previsão de 1 a 4 mm de chuva. Isto sugere que RITA considera, depois de observar apenas uma semana de variações climá-

ticas, que uma elevada temperatura mínima tende a conduzir a uma pluviosidade média (ou seja, entre 1 e 4 mm) no dia seguinte, enquanto que um mínimo baixo tanto pode conduzir a pouca como a muita chuva no dia que se segue. Isto parece surpreendente, pelo menos para mim.

Não há muito a dizer no que se refere aos valores relativos à temperatura máxima (1.9, 1.8 e 2), pelo que talvez não tenha sido muito importante. A quantidade de chuva caída, no entanto, é considerada uma variável importante. Uma fraca pluviosidade (um valor de .5) sugere pouca chuva no dia seguinte, o que concorda com o que nós próprios pensamos sobre a sucessão de um grupo de dias secos, depois uma série de dias chuvosos, etc, em vez de cada dia ser um acontecimento isolado. No entanto, em vez de dizer «muita chuva hoje significa muita chuva amanhã», RITA concluíu que um elevado valor de pluviosidade no dia de hoje (3.2) sugere uma pluviosidade média amanhã (1 a 4 mm), e que um valor moderado de queda de chuva (1.4) sugere muita chuva no dia seguinte. Depois de pensar um pouco, parece-me que isto é razoável. Se cai muita chuva hoje, talvez fique pouco para cair amanhã...

Finalmente observamos as horas de luz solar, e vemos as conclusões tiradas por RITA. Os valores apresentam aqui uma extensa variação (convindo ter presente que foram multiplicados por 16 antes de serem guardados na base de regras; a *variação* numa dada categoria é o aspecto mais importante, e não o valor absoluto). Para prever um dia com mais de 4 mm de chuva, RITA procura muito sol (a base de regras indica 35.4), enquanto que pouco sol (.8) indica 1 a 4 mm, e um pouco mais (3.8) indica que o dia será quase seco (menos de 1 mm de pluviosidade).

A base de regras de RITA, neste ponto, parece conter as seguintes regras:

— Quando se procura um dia razoavelmente húmido (acima de 4 mm de chuva), espera-se uma baixa temperatura mínima, uma pluviosidade média e bastante luz solar no dia anterior;
— Para prever um dia seco (menos de 1 mm de pluviosidade), vemos que RITA espera encontrar uma baixa temperatura mínima (se bem que ligeiramente superior ao caso anterior), pouca chuva e — o que é surpreendente — uma relativa luminosidade solar.
— No caso intermédio (1 a 4 mm de chuva), RITA procura uma temperatura mínima razoavelmente alta, uma elevada quantidade de chuva caída (relativamente aos dois outros casos) e pouca luminosidade solar.

Qual o valor destes resultados?

Se bem que seja fácil (e muito interessante) interpretar a base de regras que RITA construíu até agora, esta não tem qualquer valor se não permitir ao programa prever eficazmente o tempo. Executei o programa

durante mais dezanove dias, e registei a sua primeira previsão e, a segunda quando a dava (só apresenta uma segunda previsão, e talvez mesmo uma terceira, se os dados não apontam claramente para a primeira).

Vejamos os resultados (ao fim de 19 dias):

— Primeira previsão correcta (única) — 7 dias
— Segunda previsão correcta (se a primeira estava errada) — 7 dias
— Primeira ou segunda previsões correctas — 14 dias
— Completamente errado (primeira e segunda previsões) — 5 dias

São resultados aceitáveis, tendo em conta as descobertas «mais provável» e «seguinte mais provável». Não esqueçamos que não demos quaisquer regras a RITA. Limitámo-nos a introduzir os dados, e indicámos se a previsão feita pelo programa estava certa ou errada. Foi RITA que construíu as regras, e que a partir de um número bastante reduzido de exemplos conseguiu descobrir o suficiente sobre o tempo para fazer previsões aceitáveis. É extraordinário (pelo menos para mim) que apesar de o programa não «saber» o que está a fazer, seja capaz de criar uma regra que dê resultados na vida real.

Que aconteceria se tivéssemos dado a RITA mais dados sobre os quais pudesse trabalhar, ou nos limitássemos a pedir-lhe que indicasse se o dia seguinte ia ser seco ou húmido? Teria RITA dado melhores resultados? Tentaremos responder a estas duas perguntas com o exemplo da previsão de tempo em Melbourne.

RITA continuou a aprender durante o período de 26 dias para o qual se introduziram dados. Decidi observar todo o período a fim de verificar se os resultados tinham melhorado.

Vejamos os resultados para o mesmo período de 19 dias já referido antes:

— Primeira previsão correcta — 9 dias
— Segunda previsão correcta — 9 dias
— Primeira ou segunda previsões correctas — 17 dias
— Completamente errado (primeira e segunda previsões) — 2 dias (menos 3)

Fiquei bastante satisfeito com esta melhoria. Permite concluir que, à medida que acrescentamos exemplos (em vez de repetirmos os mesmos), RITA continua a aprender. Uma maneira de obrigar RITA a fazer bastante exercício (em vez de sermos nós a introduzir grandes quantidades de dados) consistiria em introduzir os resultados registados num «array», juntamente com os resultados correctos, deixando a RITA o trabalho de escolha de dias aleatoriamente, durante uma hora por exemplo, melhorando continuamente as suas capacidades de previsão.

Modificações

Rita define, como vimos, um número que é guardado por cada pergunta de discriminação para cada opção de saída. Se o sistema dá uma resposta errada, altera os números guardados para essa opção. Multiplica o valor guardado por cinco, soma-lhe o novo valor, e divide por seis, guardando finalmente o resultado deste cálculo. Garante assim que um acontecimento pouco usual (como um gato que ladre vigorosamente contra os ladrões, ou um dia com uma pluviosidade cem vezes superior à normal) não destrua completamente os resultados mais habituais (mas se a primeira entrada corresponder a um acontecimento invulgar, RITA demorará bastante tempo a corrigi-la).

Tentei outras relações entre dados antigos e novos (como nove vezes os antigos, mais os novos, divididos por dez; e duplicação dos antigos, mais os novos, divididos por três) mas nesses casos ou o sistema aprendia com demasiada lentidão, ou a sua base de regras flutuava excessivamente em resposta ao último grupo de valores introduzido. Como o resto deste programa (e outros sistemas periciais), este aspecto foi construído por experiência.

Dissecando RITA

Vou agora apresentar o programa RITA em algum detalhe, o que não faço geralmente neste livro. Faço-o no caso de RITA porque é o programa mais útil da obra e aquele que o leitor mais provavelmente desejará adaptar para criação dos seus próprios sistemas. Como está, é de facto um sistema de uso geral.

Sugeri já que o leitor pode tentar alterar algumas das coisas que o programa faz, quando o aplicar a domínios de sua escolha, a fim de tornar RITA mais eficaz em cada domínio específico. Utilize-o tal como está no campo que lhe interessa, e depois tente descobrir quantas respostas «certas» produz.

O programa começa por dimensionar um certo número de «arrays»:

```
1370 REM **********
1380 REM INICIALIZAÇÃO
1390 CLS
1400 REM REDUZIR ARRAYS NA LINHA SEGUINTE EM FUNÇÃO
        DAS NECESSIDADES
1410 DIM A$(50), B(50,50), C(50), D(50), E$(50), F(50), G(50)
1420 X$='' ''
1430 PRINT ''CARREGUE NUMA TECLA E EM RETORNO SE''
1440 PRINT ''DESEJA OBSERVAR A BASE DE CONHECIMENTOS''
1450 PRINT ''ACTUALIZADA APÓS CADA EXECUÇÃO; CARREGUE''
```

```
1460 PRINT ''APENAS EM RETORNO SE NÃO QUISER''
1470 INPUT U$
1480 CLS
1490 RETURN
```

Os valores de 50 dos arrays dimensionados são bastante superiores aos que provavelmente necessitará e gastam bastante memória (geralmente 12 K no meu sistema, um PC IBM). O array A$ guarda os nomes dos resultados, o E$ os nomes das perguntas de discriminação, sendo fácil reduzi-los se se conhecer à partida quantos são os resultados possíveis, e o número de perguntas que serão feitas. Podem-se alterar todos os outros valores de 50 de modo a serem iguais ao número de perguntas de discriminação feitas.

Alternativamente, o leitor poderá querer modificar o programa de tal modo que pergunte inicialmente quantos são os resultados e perguntas a que iremos recorrer, dimensionando a seguir os arrays de acordo com as respostas dadas. Esta secção do programa obtém igualmente o valor de U$ (vazia ou não), o qual determinará se o programa imprime ou não a base de regras actualizada depois de cada execução.

Depois disto RITA aceita os nomes das conclusões a que pode chegar:

```
1160 REM OPÇÕES DE SAÍDA
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
1200 PRINT ''INDIQUE NÚMERO DA OPÇÃO DE SAÍDA '';TT;''
     (CARREGUE EM RETORNO PARA TERMINAR)''
1210 INPUT A$(TT)
1220 IF A$(TT)='' ''OR TT=51 THEN TT=TT-1: RETURN
1230 GOTO 1180
```

Serão depois pedidas as perguntas de discriminação:

```
1250 REM PERGUNTAS DE DISCRIMINAÇÃO
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT ''       INDIQUE NÚMERO DA PERGUNTA'';DQ;''
    (CARREGUE EM RETORNO PARA TERMINAR)''
1340 INPUT E$(DQ)
```

```
1350 IF E$(DQ)='' '' OR DQ=51 THEN DQ=DQ-1:
     RETURN
1360 GOTO 1310
```

A partir da linha 140 RITA faz-nos algumas perguntas:

```
140 REM PERGUNTAS AO UTILIZADOR
150 CLS
160 PRINT ''EIS OS TEMAS SOBRE OS QUAIS POSSO DISCRIMINAR:''
170 PRINT
180 FOR J=1 TO TT
190 PRINT ''>'';A$(J)
200 NEXT J
210 GOSUB 1500
220 IF X$='' '' THEN PRINT ''PENSE NUM, E CARREGUE EM RETORNO''
230 IF X$<>'' '' THEN PRINT ''ESTOU PRONTA A DETERMINAR QUAL É''
240 IF X$='' '' THEN INPUT J$
250 ADD=.5
260 FOR J=1 TO DQ
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X$<>'' ''AND TT>2 THEN 390:REM
    VERIFICAR SE PERGUNTA PODE SER IGNORADA
300 PRINT ''INDIQUE UM NÚMERO ENTRE''
310 PRINT ''1 (VERDADEIRO) E O (FALSO)    ($ PARA
    TERMINAR EXECUÇÃO)''
320 PRINT:PRINT E$(J);
330 INPUT H$:IF H$=''$'' THEN PRINT:PRINT ''OBRIGADO'':
    PRINT:END
340 C(J)=VAL(H$)
350 C(J)=ADD*C(J)
360 NEXT J
370 RETURN
```

É introduzida uma cadeia (linha 330) depois de a pergunta ser impressa pela linha anterior. Se é introduzido o sinal de cifrão, o programa termina (depois de um educado OBRIGADO) na linha 330. Se não, o valor da cadeia é tornado igual a um elemento do array C (linha 340), sendo multiplicado pela variável ADD na linha seguinte. Antes de se iniciar o ciclo J (que aceita a entrada do utilizador), ADD é tornado igual a .5, sendo somado a si mesmo (ou seja, o seu valor duplica) de cada vez que o ciclo é executado (levando-o a valer 1, 2, 4, 8, 16, etc) antes de C(J) ser multiplicado por ele. Depois de o programa ter executado o ciclo em causa, o controlo volta a um ciclo de chamadas sem subrotinas perto do início do programa.

A linha 290 envia o programa para a subrotina que se inicia na linha 390, que verifica se pode ser ignorada uma pergunta:

```
390 REM VERIFICAR SE A PERGUNTA PODE SER IGNORADA
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUMP=0
430 NEXT W
440 IF JUMP=0 THEN 300
450 C(JUMP)=B(W,J)
460 GOTO 360
```

Procede por comparação de todos os valores guardados no array B para a pergunta considerada. Se estiverem a menos de .7 entre si, RITA considera que a informação que lhe corresponde pode ser ignorada. Este valor de .7 pode evidentemente ser alterado pelo leitor.

Depois de terem sido feitas todas as perguntas, RITA passa à rotina mais importante, a partir da linha 480, onde é tomada a decisão e — em, caso de necessidade — se actualiza a base de regras:

```
480 REM TOMAR DECISÕES
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXT J
520 ADD=.5
530 FOR J=1 TO TT
540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM JOGAR COM OS VALORES DAS TRÊS LINHAS SEGUINTES
    PARA OBTER RESULTADOS MAIS EFICAZES
570 IF C( X)= B(J,X) THEN D(J)=D(J)+1
580 IF ABS (C(X)-B(J,X))<.6*ADD THEN E(J)=E(J)+.4
590 IF ABS(C(X)-b(J,X))<1.2*ADD THEN F(J)=F(J)+.1
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
690 REM ** ANUNCIAR RESULTADO **
700 PRINT
710 CFLG=0
```

```
720 PRINT ''O RESULTADO MAIS PROVÁVEL É'';A$(A1)
730 IF A2<>A1 THEN PRINT ''O RESULTADO MAIS PROVÁVEL
    A SEGUIR É '';A2$(A2):CFLG=1
740 IF A3<>A2 AND A3<>A1 THEN PRINT ''O RESULTADO
    MAIS PROVÁVEL A SEGUIR É '';A$(A3):CFLG=2
750 PRINT
760 PRINT ''O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
                                    (S OU N)''
770 INPUT F$
780 IF F$<>''S'' AND F$<>''N'' THEN 770
790 IF F$=''S'' AND X$<>'' ''THEN RETURN
800 IF F$=''S'' THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 980
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CFLG=0 THEN 890
840 PRINT ''O SEGUNDO RESULTADO MAIS PROVÁVEL ESTÁ
    CORRECTO                          (S OU N)'';
850 INPUT F$
860 IF F$=''N'' THEN 890
870 IF CFLG=1 THEN A1=A2: GOTO 980
880 IF CFLG=2 THEN A1=A3: GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;''- '';A$(J)
920 NEXT J
930 PRINT
940 PRINT ''QUAL É O CORRECTO'';
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM ** EDUCAR RITA **
         (ACTUALIZAR BASE DE CONHECIMENTOS)
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+5*B
    (A1,J))/6
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(A1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$='' ''THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K);B(J,K)
```

```
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
```

O processo começa por definir os elementos dos arrays D, E e F, passando-os ao valor zero. O array D guardará os resultados «mais prováveis», E é usado para os «mais prováveis seguintes» e F para os terceiros resultados mais prováveis. Em seguida passa-se a variável ADD (usada na rotina «perguntar ao utilizador» para multiplicar a informação dada por este) para .5 de modo a poder ser usada no ciclo seguinte.

O fluxo do programa é agora controlado por um par de ciclos. O ciclo J actua entre 1 e TT (o número de resultados), e o ciclo X entre 1 e DQ (o número de perguntas de discriminação).

As três linhas que se seguem são as mais importantes do programa. É nelas que RITA toma as suas decisões. A linha 570 procura uma correspondência exacta entre a resposta introduzida, C(X), e o elemento do array B que se relaciona com o resultado possível (o valor que TT tem nesse ponto) e a pergunta para a qual C(X) é a resposta (o valor que DQ tem nesse ponto). Se encontra uma comparação exacta, D(J) é incrementada de um. Talvez o leitor verifique que o sistema funciona melhor se, exemplo, somar 1.5 ou 2 neste ponto.

A linha 580 procura uma semelhança entre um valor da base de dados e a resposta dada, e se encontra uma proximidade de .6 na resposta soma .4 a E(J). Note-se que no caso de igualdade é somado 1 a D(J), e .4 a E(J) se o valor é quase igual. A linha 590 procura uma semelhança menos próxima e, se a encontra, soma .1 a F(J).

As linhas 620 e 630 definem as variáveis A1, A2, A3, F1, F2 e F3 como iguais a zero, antes de o ciclo J (linhas 640 e 680) ser activado. Ao percorrer este ciclo, RITA passa cada variável F para o maior valor possível (passando F1 para o maior valor D(J); F2 para o maior valor E(J) e F3 para o maior valor F(J)). De cada vez que F1, F2 ou F3 são alterados, A1, A2 e A3 são-no igualmente de modo a conterem o *número* do elemento que provocou a alteração (isto é, são passados para o valor de J em causa). Isto permite a RITA dispor de um registo do elemento que se verificou ter o maior valor.

Depois de RITA ter percorrido os ciclos, A1 é igualada ao resultado que é provavelmente mais verdadeiro (porque foi atribuído a esse resultado o maior número de semelhanças entre a base de regras e as respostas do utilizador).

É chegado o momento de RITA anunciar a sua conclusão. A linha 710 passa a zero uma variável chamada CFLG («Flag de resultado próximo»). Como A$ contém os nomes dos resultados, A$(A1) é o elemento desse array que contém o nome do resultado mais provável. A linha 720 anuncia esta conclusão.

Se A2 não é igual a A1 (isto é, se a «resposta mais provável a seguir» não é igual à primeira), RITA apresenta também este resultado, e passa CFLG para 1 (valor que será usado em seguida). Se A3 possui ainda um valor diferente, é indicado um «terceiro resultado mais provável» e CFLG é passado a 2.

A linha 760 pergunta se a conclusão de RITA está correcta. Se a resposta é sim (ou seja, F$ é igualada a «S») e X$ não é igual a " "(o que acontece quando RITA sai do modo «aprendizagem» e passa ao modo «execução»), a base de regras não é modificada. Se RITA está em modo «execução» e a resposta é correcta, a base de regras não deve obviamente ser alterada. Se RITA está ainda a aprender, a linha 800 envia a acção para a rotina que se segue a 980, que actualiza a base de regras.

Dois resultados

Se só existem dois resultados, a variável TT será igual a 2. O computador chega à linha 870 depois de ter sido respondido um «N» (significando que a sua resposta estava errada). Portanto, como só há dois resultados possíveis, o outro está necessariamente correcto. Se A1 é igual a 1, então RITA deu — incorrectamente — o resultado 1 como resposta correcta. A linha 810 altera isto de tal modo que A1 passa agora a indicar a resposta correcta (ou seja, é igual a 2) antes de a acção passar a 980 a fim de actualizar a base de regras. A linha 880 faz o contrário, passando um 2 incorrecto para 1.

Se existem mais de dois resultados, RITA enfrenta um problema maior. Não será imediatamente evidente qual dos resultados restantes poderá estar correcto. A linha 830 verifica a variável CFLG, e se descobre que esta é igual a zero sabe que RITA não indicou qualquer outro «resultado mais provável», pelo que passa para a rotina que vai de 890 a 960, que pergunta ao utilizador qual a resposta correcta.

Se RITA atribuiu valores a A2 e a A3 (os «mais prováveis a seguir») diferentes do valor atribuído a A1 (o mais provável), o programa pergunta se o resultado certo é o segundo, A$ (A2). Se assim for, o valor de CFLG indica qual a resposta que foi dada como segunda escolha (se CFLG é igual a 1, é o valor de A2; se CFLG é igual a 2, é o valor de A3), pelo que RITA sabe qual a resposta correcta e passa à rotina 980 armada com esta informação, a fim de actualizar a base de regras.

A rotina que se segue, entre 900 e 960, já foi estudada anteriormente. Imprime todos os resultados possíveis no visor, e pede ao utilizador que indique qual corresponde à resposta correcta.

RITA pode agora actualizar a sua base de regras. Sabe qual é a resposta correcta (ou A1 foi seleccionada antes, ou foi igualada à res-

posta correcta pela resposta do utilizador; em qualquer dos casos indica agora a resposta certa).

O programa passa agora pelo ciclo J entre as linha 980 e 1020. Quando encontra a porção relevante da base de dados, B(A1,J), verifica se é igual a zero. Sê-lo-á se ainda não foi registada nenhuma informação neste ponto (como acontecerá sempre no início da execução). Se B(A1,J) não é igual a zero (linha 990), o programa multiplica o valor actualmente guardado nele por cinco, soma o valor obtido na execução actual, e depois divide o total por seis. Garante-se assim que (a) toda a informação das execuções anteriores é mantida apesar desta resposta; (b) o impacto da resposta actual não é ignorado; e (c) uma resposta atípica não destrói a base de regras.

Se B(A1,J) é igual a zero (linha 1000), então este elemento é tornado igual à única resposta que o sistema encontrou até agora, ou seja, é tornado igual a A1. A linha 1010 elimina as casas decimais (sem esta linha RITA começa a guardar valores até seis casas decimais, o que é absurdo dada a natureza altamente subjectiva dos dados originais).

Finalmente, nesta secção extensa — mas muito importante — do programa, RITA verifica na linha 1040 se U$ é igual a " "; se assim acontecer, tal significa que o utilizador indicou no início da execução que não pretendia ver o estado actual da base de regras impresso. Neste caso, o programa volta ao ciclo de controlo existente no princípio do programa a fim de receber a série seguinte de entradas. Se pelo contrário o utilizador indicou de início o desejo de ver o estado da base de regras (e esta é, em minha opinião, a parte mais interessante de todo o processo), a secção que se segue imprime as informações pedidas, como se pode ver nos exemplos de execução.

LISTAGEM COMPLETA DE RITA

Várias secções do programa RITA foram já apresentadas no capítulo anterior. No entanto, não apresentámos aí toda a listagem. No capítulo que se segue utilizaremos RITA com um novo conjunto de dados, e mostraremos como se podem tratar os dados de entrada de modo a satisfazerem uma escala entre 0 e 1.

Antes disso, porém, mostramos a listagem completa do programa:

```
10 REM RITA
20 GOSUB 1380:REM INICIALIZAR
30 GOSUB 1160:REM INDICAR OPÇÕES
40 GOSUB 1250:REM OPÇÕES DE DISCRIMINAÇÃO
50 GOSUB 140:REM QUESTIONAR O UTILIZADOR
60 GOSUB 480:REM TOMAR DECISÃO E ACTUALIZAR
   BASE DE CONHECIMENTOS
70 PRINT ''CARREGUE EM RETORNO PARA CONTINUAR''
80 IF X$<>'' ''THEN INPUT I$:GOTO 50
90 PRINT ''APRENDIZAGEM''
100 PRINT ''OU QUALQUER OUTRA TECLA E RETORNO PARA USAR RITA'';
110 INPUT X$:GOTO 50
120 END
130 REM ***************************
140 REM PERGUNTAS AO UTILIZADOR
150 CLS
160 PRINT ''EIS OS TEMAS SOBRE OS QUAIS POSSO DISCRIMINAR:''
170 PRINT
180 FOR J=1 TO TT
190 PRINT ''>'';A$(J)
200 NEXT J
210 GOSUB 1500
220 IF X$='' ''THEN PRINT ''PENSE NUM, E CARREGUE EM RETORNO''
230 IF X$<>'' ''THEN PRINT ''ESTOU PRONTA A DETERMINAR QUAL É''
240 IF X$='' ''THEN INPUT J$
250 ADD=.5
260 FOR J=1 TO DQ
```

```
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X $<>'' '' AND TT>2 THEN 390:REM VERIFICAR SE PERGUNTA
    PODE SER IGNORADA
300 PRINT ''INDIQUE UM NÚMERO ENTRE''
310 PRINT ''1 (VERDADEIRO) E 0 (FALSO)  ($ PARA TERMINAR
    EXECUÇÃO)''
320 PRINT:PRINT E$(J);
330 INPUT H$:IF H$=''$'' THEN PRINT:PRINT ''OBRIGADO'':PRINT:END
340 C(J)=V AL(H$)
350 C(J)=ADDC(J)
360 NEXT J
370 RETURN
380 REM ***************************
390 REM VERIFICAR SE A PERGUNTA PODE SER IGNORADA
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUMP=0
430 NEXT W
440 IF JUMP=0 THEN 300
450 C(JUMP)=B(W,J)
460 GOTO 360
470 REM ***************************
480 REM TOMAR DECISÕES
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXT J
520 ADD=.5
530 FOR J=1 TO TT
540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM JOGAR COM OS VALORES DAS TRÊS LINHAS SEGUINTES PARA OBTER
    RESULTADOS MAIS EFICAZES
570 IF C(X)=B(J,X) THEN D(J)=D(J)+1
580 IF ABS(C(X)-B(J,X))<.6*ADD THEN E(J)=E(J)+.4
590 IF ABS (C(X)-B(J,X))<1.2*ADD THEN F(J)=F(J)+.1
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
```

```
690 REM ** ANUNCIAR **
700 PRINT
710 CFLG=0
720 PRINT ''O RESULTADO MAIS PROVÁVEL É'';A$(A1)
730 IF A2<>A1 THEN PRINT ''O RESULTADO MAIS PROVÁVEL A SEGUIR
    É'';A$(A2):CFLG=1
740 IF A3<> A2 AND A3<>A1 THEN PRINT ''O RESULTADO MAIS PROVÁVEL
    A SEGUIR É'';A$(A3):CFLG=2
750 PRINT
760 PRINT ''O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
                                        (S OU N)'';
770 INPUT F$
780 IF F$<>''S'' AND F$ <''N'' THEN 770
790 IF F$=''S'' AND X$ <>'' ''THEN RETURN
800 IF F$=''S'' THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 980
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CFLG=0 THEN 890
840 PRINT ''O SEGUNDO RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
                    (S OU N)'';
850 INPUT F$
860 IF F$=''N'' THEN 890
870 IF CFLG=1 THEN A1=A2:GOTO 980
880 IF CFLG=2 THEN A1=A3:GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;''-'';A$(J)
920 NEXT J
930 PRINT
940 PRINT ''QUAL É O CORRECTO'';
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM ** EDUCAR RITA ** (ACTUALIZAR BASE DE CONHECIMENTOS)
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+5*B (A1,J))/6
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(A1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$='' ''THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
```

```
1100 PRINT E$(K);B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
1150 REM ***************************
1160 REM OPÇÕES DE SAÍDA
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
1200 PRINT ''INDIQUE NÚMERO DA OPÇÃO DE SAÍDA'';TT;'' (CARREGUE
     EM RETORNO PARA TERMINAR)''
1210 INPUT A$(TT)
1220 IF A$(TT)='' ''OR TT=51 THEN TT=TT-1:RETURN
1230 GOTO 1180
1240 REM ***************************
1250 REM PERGUNTAS DE DISCRIMINAÇÃO
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT '' INDIQUE NÚMERO DA PERGUNTA '';DQ;'' (CARREGUE EM
     RETORNO PARA TERMINAR)''
1340 INPUT E$(DQ)
1350 IF E$(DQ)= '' ''OR DQ=51 THEN DQ=DQ-1: RETURN
1360 GOTO 1310
1370 REM ***************************
1380 REM INICIALIZAÇÃO
1390 CLS
1400 REM REDUZIR ARRAYS NA LINHA SEGUINTE EM FUNÇÃO DAS
     NECESSIDADES
1410 DIM A$(50),B(50,50),C(50),D(50),E$(50),F(50),E(50)
1420 X$='' ''
1430 PRINT ''CARREGUE NUMA TECLA E EM RETORNO SE''
1440 PRINT ''DESEJA OBSERVAR A BASE DE CONHECIMENTOS''
1450 PRINT ''ACTUALIZADA APÓS CADA EXECUÇÃO; CARREGUE''
1560 PRINT ''APENAS EM RETORNO SE NÃO QUISER''
1470 INPUT U$
1480 CLS
1490 RETURN
1500 PRINT  ''-------------------------------------''
1510 RETURN
```

9

O GABINETE DE METEOROLOGIA

Neste capítulo vamos transformar RITA em especialista no clima de Dezembro em Melbourne, Austrália. Explicarei os passos dados para criar este sistema com algum pormenor. O leitor obterá assim uma ideia clara da forma como pode adaptar RITA, de modo a criar sistemas periciais próprios.

Em primeiro lugar, é necessário dar aos dados uma forma apropriada a este programa. Dissémos já que RITA espera que as entradas variem na gama 0 a 1, correspondendo 0 a falso e 1 a verdadeiro, e equivalendo os valores intermédios a diferentes graus de verdade. Não é necessário usar apenas o zero ou o um. RITA é extremamente tolerante. No entanto, é mais simples definir uma regra e obedecer sempre a ela quando se está a desenvolver um sistema a partir do «esqueleto» de RITA, do que ser forçado a experimentar mais tarde qual a melhor escala a usar.

Os dados que vamos fornecer a RITA neste exercício provêm dos valores oficiais das condições climáticas de Dezembro de 1984. Daremos ao computador as leituras barométricas diárias às 9 horas da manhã, as temperaturas máxima e mínima diárias e a humidade relativa às 3 horas da tarde. A partir destes valores, RITA indicará se chove ou não no dia seguinte.

As autoridades oficiais indicam que o mês estudado contém o maior número de dias de chuva (14) desde 1976. Isto é bom, dado que significa que cerca de metade dos dias são húmidos, em vez de o ser apenas alguns dias, como aconteceu em 1982. Um mês completamente seco constituiria um desafio irrelevante para RITA, e pouco teria demonstrado.

Vejamos os dados correspondentes aos primeiros dias:

DATA	BAROM	TEMP MIN	TEMP MAX	HR%	PRECI-PITAÇÃO
1	1011.6	11.0	25.5	31	0
2	1006.7	12.6	27.6	29	0
3	1012.4	10.8	16.5	52	2.6

Vê-se que os números são bastante diferentes, variando as leituras barométricas à volta de 1000, as temperaturas entre 10 e 30, a percentagem de humidade relativa entre 0 e 100, e a precipitação entre zero e infinito. Como transformaremos estes valores de modo ao satisfazerem uma escala entre zero e um?

É muito simples e o seu computador fará quase todo o trabalho sozinho. Escreva e execute o programa seguinte:

```
10 REM ESCALA
20 DIM X(50),Z(50)
30 CLS
40 INPUT ''MAIOR VALOR'';A
50 INPUT ''MENOR VALOR'';B
60 A=A+.001
70 B=B-.001
80 C=(A-B)/50
90 X(0)=B
100 FOR J=1 TO 50
110 X(J)=(J-1)+C
120 Z(J)=J/50
130 PRINT Z(J),X(J)
140 NEXT J
150 DIFF=(X(2)-X(1))/2
160 COUNT=0
170 COUNT=COUNT+1
180 PRINT ''INDIQUE VALOR'';COUNT
190 INPUT Q$
200 IF Q$='' ''THEN END
210 Q=VAL(Q$)
220 IF Q<B OR Q>A THEN 180
230 FOR J=0 TO 50
240 IF ABS(Q-X(J))<DIFF THEN LPRINT COUNT;'' -'';Z(J)
250 NEXT J
260 GOTO 170
```

Execute o programa e responda aos seus pedidos. O leitor dispõe de uma lista dos valores a introduzir, por exemplo os barométricos. O computador pergunta «MAIOR VALOR?", pelo que procuramos o valor mais elevado da lista. No meu caso é 1018.1, pelo que indico este valor. Em seguida a máquina pergunta «MENOR VALOR?», ao que respondo 994.2.

O computador pede-nos agora que introduzamos os dados necessários, indicando simultaneamente o número (a variável COUNT da linha 180) do elemento da lista, para o caso de nos perdermos. Escrevemos o primeiro valor (1011.6) da lista, e a linha 240 imprime (neste

caso para a impressora; se preferir no visor escreva PRINT em vez de LPRINT) o valor .72, equivalente àquele na escala zero a um.

Introduzimos assim todos os dados relativos ao mês em causa, tomando nota (ou permitindo que o próprio computador o faça) dos resultados a fim de poderem ser introduzidos no programa RITA quando for apropriado.

Seguimos o mesmo processo para as temperaturas mínimas, as máximas e a humidade relativa. Podemos começar agora a ensinar RITA sobre as variações do tempo na Austrália. Antes do mais, no entanto, fiz algumas pequenas alterações no programa.

O leitor recorda que já foi dito que o esqueleto do programa dado é um tanto grosseiro, e que convirá modificá-lo de modo a produzir os resultados mais eficazes para cada caso.

Como a entrada deste programa é dada com duas casas decimais, pareceu absurdo reduzi-la na linha 1010 a uma única casa decimal, eventualmente perdendo informação importante. Para resolver isto, alterámos a linha 1010 para o seguinte:

```
1010 B(A1,J)=INT(100*B(A1,J))/100
```

A linha 570, que procura semelhanças entre os dados de cada execução e a base de dados, foi modificada de modo a procurar estas semelhanças do seguinte modo:

```
570 IF ABS(C(X)-B(J,X))<.2*ADD THEN D(J)=D(J)+1
```

A educação de Rita

Podemos começar a educar RITA. Começamos por dizer-lhe que as opções de saída são CHOVE AMANHÃ e NÃO CHOVE AMANHÃ, e indicamos as perguntas de discriminação a fazer:

```
--------------------------------------------------
      INDIQUE PERGUNTA NÚMERO 1
      (CARREGUE EM RETORNO PARA TERMINAR)
? BARÓMETRO
--------------------------------------------------
      INDIQUE PERGUNTA NÚMERO 2
      (CARREGUE EM RETORNO PARA TERMINAR)
? TEMPERATURA MÍNIMA
--------------------------------------------------
      INDIQUE PERGUNTA NÚMERO 3
      (CARREGUE EM RETORNO PARA TERMINAR)
? TEMPERATURA MÁXIMA
--------------------------------------------------
```

```
    INDIQUE PERGUNTA NÚMERO 4
    (CARREGUE EM RETORNO PARA TERMINAR)
? HUMIDADE RELATIVA
```

Depois de quatro dias de entradas e correcções, a base de regras de RITA passou a ser a seguinte:

```
-----------------------------------------------

CHOVE AMANHÃ

BARÓMETRO .6
TEMPERATURA MÍNIMA .7
TEMPERATURA MÁXIMA 2.46
HUMIDADE RELATIVA 2.15
-----------------------------------------------
NÃO CHOVE AMANHÃ

BARÓMETRO .72
TEMPERATURA MÍNIMA .48
TEMPERATURA MÁXIMA 2.88
HUMIDADE RELATIVA 1.12
-----------------------------------------------
```

A regra mais notável criada por RITA respeita à humidade relativa. Se for baixa, é provável que o dia seguinte seja seco. Isto parece bastante razoável. RITA parece igualmente pensar que as baixas temperaturas mínimas e as elevadas leituras barométricas apontam do mesmo modo para um dia seco.

Executei o programa durante o equivalente a mais quatro dias de introdução de dados. A base de regras de RITA alterou-se do seguinte modo:

```
-----------------------------------------------

CHOVE AMANHÃ

BARÓMETRO .6
TEMPERATURA MÍNIMA .7
TEMPERATURA MÁXIMA 2.46
HUMIDADE RELATIVA 2.15
-----------------------------------------------
NÃO CHOVE AMANHÃ

BARÓMETRO .84
TEMPERATURA MÍNIMA .55
TEMPERATURA MÁXIMA 2.12
HUMIDADE RELATIVA 2.36
```

Se bem que RITA tenha mantido as suas opiniões anteriores sobre as leituras de barómetro e as temperaturas mínimas, mudou completamente no que se refere à humidade relativa. Continuei pacientemente a introduzir dados. No final do mês, RITA definira as seguintes regras:

```
------------------------------------------------

CHOVE AMANHÃ

BARÓMETRO .43
TEMPERATURA MÍNIMA 1
TEMPERATURA MÁXIMA 2.23
HUMIDADE RELATIVA 3.74
------------------------------------------------

NÃO CHOVE AMANHÃ

BARÓMETRO .75
TEMPERATURA MÍNIMA .95
TEMPERATURA MÁXIMA 2.86
HUMIDADE RELATIVA 2.15
```

Uma baixa leitura barométrica, uma elevada temperatura mínima e uma baixa temperatura máxima, juntamente com uma elevada humidade relativa, parecem apontar para CHOVE AMANHÃ, enquanto que as condições opostas sugerem o contrário. Estas conclusões parecem razoáveis. Mas que resultado dão na prática?

No mês em causa, ignorando o primeiro dia (dado que a resposta de RITA se baseia inteiramente na ordem pela qual se introduzem as opções de resultado), há 29 dias para verificar a validade dos julgamentos do programa. RITA previu a presença ou ausência de chuva correctamente em 18 desses dias, o que parece bastante bom. Esta impressão foi reforçada ao examinar os valores, mostrando que RITA previu como seco um dia em que apenas se observou uma pluviosidade de 0,2 mm, e que alguns dias húmidos intercalados entre outros secos foram previstos adequadamente.

Para verificar se RITA continuava a aprender, executei de novo todo o mês. No final desta segunda execução, RITA tinha desenvolvido a seguinte base de regras:

```
------------------------------------------------

CHOVE AMANHÃ

BARÓMETRO .43
TEMPERATURA MÍNIMA 1.01
TEMPERATURA MÁXIMA 2.13
HUMIDADE RELATIVA 3.97
------------------------------------------------
```

```
NÃO CHOVE AMANHÃ

BARÓMETRO .75
TEMPERATURA MÍNIMA .98
TEMPERATURA MÁXIMA 2.58
HUMIDADE RELATIVA 2.22
```

Vê-se assim que RITA reforçou no essencial a sua opinião anterior, aumentando um pouco o valor da humidade relativa quando procura um dia seco. Os resultados neste segundo mês melhoraram relativamente à primeira execução. Os mesmos 29 dias produziam 19 previsões correctas, se bem que continuassem a afirmar que o dia com 0,2 mm de chuva seria seco. O programa previu correctamente, por outro lado, que o segundo dia do mês seria seco (algo que da primeira vez não acontecera), conseguindo assim 20 resultados correctos em 30.

Vejamos os resultados apresentados por RITA, sendo os erros indicados por X;

Dia	Tempo	Previsão (primeira)		Previsão (segunda)	
2	SECO	-		SECO	
3	HÚMIDO	SECO	X	HÚMIDO	
4	HÚMIDO	SECO	X	HÚMIDO	
5	HÚMIDO	HÚMIDO		HÚMIDO	
6	SECO	SECO		HÚMIDO	X
7	SECO	HÚMIDO	X	SECO	
8	SECO	SECO		SECO	
9	SECO	SECO		SECO	
10	SECO	SECO		SECO	
11	HÚMIDO	SECO	X	SECO	X
12	HÚMIDO	HÚMIDO		HÚMIDO	
13	SECO	HÚMIDO	X	HÚMIDO	X
14	HÚMIDO	HÚMIDO		HÚMIDO	
15	HÚMIDO	SECO	X	SECO	X
16	HÚMIDO	HÚMIDO		HÚMIDO	
17	SECO	SECO		SECO	
18	HÚMIDO	SECO	X	SECO	X
19	SECO	HÚMIDO	X	HÚMIDO	X
20	HÚMIDO	SECO	X	SECO	X
21	HÚMIDO	SECO	X	SECO	X
22	SECO	SECO		SECO	
23	SECO	SECO		SECO	
24	SECO	SECO		SECO	
25	HÚMIDO	HÚMIDO		SECO	X
26	HÚMIDO	HÚMIDO		HÚMIDO	

Dia	Tempo	Previsão (primeira)		Previsão (segunda)	
27	HÚMIDO	SECO	X	SECO	X
28	SECO	SECO		SECO	
29	SECO	SECO		SECO	
30	SECO	SECO		SECO	

Uma tabela como esta é certamente muito útil quando se tenta produzir um sistema pericial, dado que ilustra os erros cometidos. Por exemplo, em ambas as execuções os dias 18.º e 21.º estiveram errados. Conviria descobrir quais os valores dados nesses dias, a fim de verificar se a correcção de alguns pormenores do programa permitiria aumentar as suas capacidades de discriminação.

O leitor talvez deseje agora utilizar algumas estatísticas diárias relativas à sua própria cidade, verificando a qualidade do trabalho produzido por RITA.

Novas entradas

Se quiser, pode modificar a secção do utilizador de tal modo que em vez de introduzir um número entre zero e um, o utilizador se limite a seleccionar uma palavra de menú, que é depois traduzida internamente por um número apropriado ao sistema.

Um tal menú (com o número que RITA cria a partir da resposta dada) pode ser do seguinte tipo:

```
SELECCIONE A OPÇÃO QUE É VERDADEIRA PARA ESTA QUESTÃO:

A - SEMPRE                                        (1)
B - GERALMENTE                                    (.8)
C - METADE DAS VEZES                              (.6)
D - ALGUMAS VEZES                                 (.4)
E - RARAMENTE                                     (.2)
F - NUNCA                                         (0)
```

Talvez isto torne a sua RITA não só mais fácil de usar como ainda mais eficaz, dado que tenderá a obter respostas mais coerentes do utilizador, do que no caso de estar dependente de uma estimativa realizada por este em cada momento.

10

LÓGICA E PROGRAMAÇÃO

Levar uma máquina a funcionar logicamente é um passo vital na tentativa de mimar um comportamento «inteligente». Os que tentam programar o comportamento lógico possuem atrás de si uma longa história de estudo da lógica.

O famoso silogismo de Aristóteles:

TODOS OS HOMENS SÃO MORTAIS
ARISTÓTELES É UM HOMEM
PORTANTO ARISTÓTELES É MORTAL

introduziu um conceito lógico fundamental: as conclusões decorrem de permissas bem definidas.

Infelizmente, os computadores não são integrados neste tipo de pensamento pelas linguagens de programação mais usadas actualmente. A maior parte das linguagens de computador, incluindo a BASIC, são *imperativas.* Ou seja, são construídas quase exclusivamente por comandos que devem ser obedecidos pelo computador (LET X=95:LET Y=2*X:PRINT Y). Uma linguagem deste tipo não é a melhor para escrever programas que mimem o pensamento lógico.

Linguagens declarativas

Para tal necessitaremos de considerar as linguagens de programação declarativas. Estas constroem programas a partir de definições, que descrevam relações entre elementos que o computador está a manipular. Quando é executado um programa imperativo, o computador segue um certo número de ordens, tomando decisões do tipo IF/THEN, e produzindo na saída os resultados do processamento. Quando executa um programa declarativo, o computador utiliza as definições que satisfazem uma dada relação entre os elementos considerados. A saída de um tal programa é a relação que revela.

A maioria das linguagens de computador actualmente usadas, como a BASIC e a FORTRAN, dão óptimos resultados quando a ta-

refa a executar é de tipo «linear», quando o método usado exige um «polícia» (a unidade de processamento central) para dirigir o «tráfego de pensamento» por um caminho bem definido. Mas este método não é mais apropriado para as exigências da inteligência artificial ou para os sistemas periciais onde é necessário que diversos elementos interactuem simultaneamente e livremente.

O trabalho realizado por organizações como o Instituto Japonês de Tecnologia de Computadores da Nova Geração ou o Programa Alvey em Inglaterra decorre em linha recta do trajecto correspondente aos métodos de solução de problemas em computador definidos por von Neumann, que é seguido desde a década de 1940. O computador de quinta geração, em vez de conter um único processador trabalhando sequencialmente, dispõe de um certo número de processadores trabalhando em paralelo, cada um deles dedicado a uma tarefa separada (se bem que todas estejam relacionadas). Cada uma destas tarefas se assemelha a uma subrotina de um programa principal, exceptuando o facto de, em vez de serem chamadas uma a uma, e só em determinadas fases da execução do programa, serem executadas simultaneamente enviando e recebendo informações entre si.

O trabalho das equipas Alvey e japonesa concentra-se, em parte, no uso de linguagens de programação descritivas como a LISP (LISt Processing) e das suas derivadas, como a PROLOG (PROgramming in LOGic) e Logo. Examinaremos a LISP e a PROLOG (juntamente com duas outras linguagens mais simples — a EASLE e a HASTE — que desenvolvi como introduções ao uso das linguagens descritivas ou declarativas) nesta secção, e quando o leitor chegar ao fim encontrará versões de cada uma delas que poderá utilizar no seu computador.

LISP

A história da LISP iniciou-se em 1956, quando se realizou o primeiro seminário sobre inteligência artificial no Dartmouth College. Foi organizado por quatro homens, incluindo um jovem assistente de matemáticas, John McCarthy. Os quatro apresentaram uma proposta à Fundação Rockefeller sobre a realização de uma conferência tratando da possibilidade de quaisquer características da inteligência poderem ser descritas com exactidão suficiente para permitir a sua simulação pela máquina (McCorduck, 1979).

Um dos documentos desta conferência, apresentado por Herbet Simon, tratava de uma linguagem de processamento de listas um tanto deselegante por ele desenvolvida e chamada IPL (Information Processing Language). Chris Bidmead, na revista Practical Computing de Outubro, 1984 (pág. 129), indica que a IPL e esse documento constituiram a semente que veio a dar origem à LISP. «O seu (da IPL)

pseudocódigo de baixo nível e a sua sintaxe semelhante à de um assemblador sugeriram-lhe (a McCarthy) a ideia de uma linguagem algébrica de processamento de listas...».

McCarthy usou as ideias de Simon (juntamente com as de diversas outras pessoas que trabalhavam neste campo, incluindo Alan Newell, J. C. Shaw e Gelernter da IBM) para o desenvolvimento da LISP. A primeira versão começou a funcionar em 1958 (LISP 1), e a partir dela produziu a LISP 1.5, principal antecessora da maioria das versões desta linguagem actualmente em uso, incluindo evidentemente o programa SSLISP apresentado no final desta secção.

A LISP começou com dois tipos de dados — átomos e listas. As listas são formadas por átomos e outras listas. A LISP não utiliza verdadeiramente programas enquanto tais, limitando-se a avaliar listas. Dados e programas são portanto, em grande medida, impossíveis de distinguir em LISP. Usa-se um programa LISP pedindo à linguagem que consulte a sua base de dados à procura de uma lista (ou átomo) que satisfaça certas condições.

PROLOG

A PROLOG, a mais vigorosa «cria» da LISP, faz um uso considerável da falta de distinção entre dados e programas. Um programa PROLOG é essencialmente formado por uma base de dados contendo listas que podem ser interrogadas.

A linguagem foi inventada por Alain Colmerauer em inícios da década de 1970, tendo sido primeiramente instalada em Marselha em 1972 por Colmerauer e Roussel, sob a forma de um interpretador escrito em Algol-W. Foi reescrita no ano seguinte usando a Fortran. A nova versão era consideravelmente mais eficaz e depressa se espalhando pelo mundo académico na Europa e nos Estados Unidos. As universidades e os investigadores de inteligência artificial desenvolveram gradualmente as suas próprias versões da linguagem na década que se seguiu à sua primeira implementação. A PROLOG da Universidade de Edimburgo, instalada num DEC-10, que foi a primeira a incorporar um compilador, é geralmente considerada como a versão «standard» da linguagem.

A popularidade da PROLOG aumentou consideravelmente desde que os japoneses anunciaram a sua intenção de usá-la como elemento principal do seu projecto de computadores de quinta geração.

Micro-PROLOG

Existem hoje versões da PROLOG para muitos microcomputadores. A primeira destas, a micro-PROLOG, foi escrita por Frank G. McCabe e Keith L. Clark, do Imperial College em Londres, na Uni-

dade de Programação Lógica. Apareceu em 1982, em assemblador Z80 para sistemas CP/M 2.2. Existe hoje para muitos computadores, correndo em MS-DOS ou CP/M-86. A micro-PROLOG possui uma sintaxe muito mais simples do que a de outras versões, como a DEC-10 de Edimburgo. No entanto, a linguagem pode ser facilmente ampliada pelo utilizador, sendo suficientemente potente para permitir a realização de trabalho bastante útil.

A micro-PROLOG inclui um programa «front-end» chamado SIMPLE, mais fácil de utilizar do que a estrutura de listas da linguagem. Mais adiante apresentarei ao leitor um programa que emula a SIMPLE em micro-PROLOG, a fim de lhe permitir aprender alguma coisa sobre a linguagem sem ter de gastar dinheiro a adquiri-la. Depois poderá decidir se o seu interesse pela PROLOG é suficientemente forte para justificar a compra de um compilador.

Um programa PROLOG é formado por uma base de dados contendo os factos a investigar e as regras para o fazer. Para facilitar ao leitor a «entrada» suave nas linguagens de tipo declarativo ou descritivo, decidi inventar uma linguagem do mesmo tipo, obviamente bastante primitivo. Será apenas necessário ao leitor introduzir na sua própria máquina um programa relativamente curto para poder utilizar a linguagem em causa. A habilidade que conseguir adquirir na utilização prática desta linguagem — que é apresentada no capítulo que se segue — poderá depois ser aplicada no uso da versão simplificada da SIMPLE que apresentaremos mais adiante e que constituirá uma melhor introdução à PROLOG.

11

PENSANDO EM «HASTE»

A minha linguagem é chamada HASTE (de «HArtnell's Simple declarative TonguE»...). Construimos uma base de dados em HASTE escrevendo frases que contêm um asterisco que de facto as divide em sujeito e predicado. O computador aceita estas frases, e a partir delas pode responder a perguntas e chegar a conclusões. Isto é fácil de compreender se observarmos o seguinte exemplo de execução em HASTE. Antes do mais, indicamos alguns factos ao computador:

```
>JOÃO*É UM HOMEM
>PEDRO*TEM MEDO DO LOBO
>MARIA*TEM MEDO DO LOBO
>PEDRO*É UM HOMEM
>MARIA*É UMA MULHER
>PEDRO*SOBE ÀS ÁRVORES
>MARIA*SOBE ÀS ÁRVORES
>UM REMENDO A TEMPO*POUPA NOVE
>TOSTÃO POUPADO*É TOSTÃO GANHO
>PEDRO*TEM DOIS METROS DE ALTURA
>PEDRO*TEM QUATORZE ANOS
>PEDRO*É ESPECIALISTA EM COMPUTADORES
>MARIA*É ESPECIALISTA EM COMPUTADORES
>MARIA*TEM MAIS DE DOIS METROS DE ALTURA
```

Note-se que o asterisco se encontra a meio da frase, directamente antes do verbo e ocupando a posição de um espaço.

Para interrogar a base de dados escreve-se um ponto de interrogação depois de aparecer o sinal>. Se desejamos verificar se um determinado facto está guardado em HASTE, limitamo-nos a acrescentar ao sinal de interrogação a instrução que pretendemos verificar. O programa responde com verdadeiro ou falso, e depois escreve a linha FIM DE RESPOSTA a fim de mostrar que a informação dada constitui tudo o que tem a dizer quanto à pergunta feita.

Experimentemos verificar se HASTE aprendeu alguns factos:

```
>?PEDRO*É UM HOMEM
VERDADEIRO
        >FIM DE RESPOSTA<
>?PEDRO*TEM MEDO DO LOBO
VERDADEIRO
        >FIM DE RESPOSTA<
```

Se quisermos saber o que HASTE conhece sobre um dado tema, substituímos o que não interessa pelo sinal «/». Neste caso, HASTE revela tudo o que sabe sobre a parte indicada (no caso seguinte, é como se perguntássemos ao programa: «Diz-me tudo o que sabes sobre Pedro»):

```
>?PEDRO*/
TEM MEDO DO LOBO
É UM HOMEM
SOBE ÀS ÁRVORES
TEM DOIS METROS DE ALTURA
TEM QUATORZE ANOS
É ESPECIALISTA EM COMPUTADORES
        >FIM DE RESPOSTA<
```

Podemos também indicar a parte da frase correspondente ao predicado, e HASTE responderá todos os sujeitos que lhe correspondem:

```
>?/*TEM MEDO DO LOBO
PEDRO
MARIA
        >FIM DE RESPOSTA<
>?UM REMENDO A TEMPO*/
POUPA NOVE
        >FIM DE RESPOSTA<
```

Espero que o leitor tenha já uma ideia, apesar das limitações da linguagem, sobre a forma como podem ser interrogadas as linguagens declarativas, e como podem formar sistemas periciais com um imenso poder.

Existe uma forma de interrogar muito mais útil do que as anteriormente citadas: aquela em que o computador deve verificar a verdade de duas afirmações, e fornecer uma informação que satisfaça ambas as condições. A pergunta que se segue, e que utiliza a fun-

ção AND, pergunda a HASTE «Quais os sujeitos que têm medo do lobo e são homens»:

```
>?/*TEM MEDO DO LOBO AND /*É UM HOMEM
PEDRO
        >FIM DE ESPERA<
```

Ou ainda, «Quais os sujeitos que sobem às árvores e são especialistas em computadores?»:

```
>?/*TEM MEDO DO LOBO AND /*É ESPECIALISTA EM COMPUTADORES
PEDRO
MARIA
        >FIM DE RESPOSTA<
```

No caso de algumas perguntas não há resposta possível:

```
>?/*SOBE ÀS ÁRVORES AND /*É TOSTÃO GANHO
        >FIM DE RESPOSTA<
```

Se HASTE fosse, por exemplo, um sistema pericial em medicina, poderíamos perguntar-lhe «Que doença provoca manchas avermelhadas e aparece em crianças com quatro anos de idade?». A informação que o sistema pericial HASTE utilizaria para responder teria sido introduzida em linguagem natural (exceptuando o asterisco). Esta é uma das vantagens das linguagens declarativas. Permitem o uso da linguagem natural (dentro de certas restrições, como é óbvio) nas entradas e podem responder de forma compreensível.

Se quisermos descobrir tudo aquilo que o sistema conhece, escrevemos o sinal «/» de cada lado do asterisco:

```
>?/*/
JOÃO*É UM HOMEM
PEDRO*TEM MEDO DO LOBO
MARIA*TEM MEDO DO LOBO
PEDRO*É UM HOMEM
MARIA*É UMA MULHER
PEDRO*SOBE ÀS ÁRVORES
MARIA*SOBE ÀS ÁRVORES
UM REMENDO A TEMPO*POUPA NOVE
TOSTÃO POUPADO*É TOSTÃO GANHO
PEDRO*TEM DOIS METROS DE ALTURA
PEDRO*TEM QUATORZE ANOS
PEDRO*É ESPECIALISTA EM COMPUTADORES
MARIA*É ESPECIALISTA EM COMPUTADORES
MARIA*TEM MAIS DE DOIS METROS DE ALTURA
        >FIM DE RESPOSTA<
```

Vejamos a resposta dada pelo sistema a mais algumas perguntas:

```
>?MARIA*/
TEM MEDO DO LOBO
É UMA MULHER
SOBE ÀS ÁRVORES
É ESPECIALISTA EM COMPUTADORES
TEM MAIS DE DOIS METROS DE ALTURA
        >FIM DE RESPOSTA<
>?JOÃO*/
É UM HOMEM
        >FIM DE RESPOSTA<
>?PEDRO*/
TEM MEDO DO LOBO
É UM HOMEM
SOBE ÀS ÁRVORES
TEM DOIS METROS DE ALTURA
TEM QUATORZE ANOS
É ESPECIALISTA EM COMPUTADORES
        >FIM DE RESPOSTA<
```

Antes de apresentarmos a listagem de HASTE, para que o leitor possa experimentar a linguagem por si mesmo, vejamos um resumo das suas regras de funcionamento:

1 — Todas as entradas são dadas sob a forma de uma frase, com um asterisco entre o sujeito e o predicado.
2 — Interroga-se a base de dados precedendo a pergunta por um ponto de interrogação.
3 — Para verificar se HASTE conhece um dado facto, escreve-se este, precedido por um ponto de interrogação. O sistema responderá VERDADEIRO (significando que o conhece) ou FALSO (no caso contrário).
4 — Usa-se um traço na diagonal, noutras perguntas, em substituição da parte das frases que se deseja conhecer. Isto significa que ?/*PAI DE TOMÁS produzirá algo do tipo JOÃO É. ?/* provocará a impressão de toda a base de dados; e ?JOÃO É*/ imprimirá algo do tipo PAI DE TOMÁS.
5 — A base de dados pode igualmente responder a perguntas que utilizem a função AND, apresentando as respostas para as quais ambas as condições são verdadeiras; portanto ?JOÃO/* AND PAI/* produzirá toda a informação verdadeira para JOÃO e para PAI.

Como o leitor compreenderá ao examinar a listagem, HASTE produz estes resultados muito simplesmente manipulando os elementos

de um par de arrays. Podemos guardar até 255 factos na base de dados. Eis a listagem:

```
10 REM HASTE
20 DIM A$(255),B$(255)
30 F=0:CLS
40 REM
50 FLAG=0
60 INPUT''>'',D$
70 IF D$='' '' THEN END
80 IF LEFT$(D$,1)=''?'' THEN 200
90 E=0
100 E=E+1
110 IF MID$(D$,E,1)=''*''THEN 140
120 IF E<LEN(D$) THEN 100
130 PRINT ''ENTRADA INCORRECTA'': GOTO 50
140 IF FLAG=3 THEN RETURN
150 F=F+1:IF F=256 THEN END
160 A$(F)=LEFT$(D$,E-1)
170 B$(F)=MID$(D$,E+1)
180 GOTO 50
190 REM
200 REM INTERROGAR
210 FLAG=4:TRUE=0
220 IF RIGHT$(D$,1)=''/'' THEN FLAG=3
230 FOR J=1 TO LEN(D$)-5
240 IF MID$(D$,j,5)='' AND '' THEN FLAG=5 : TRUE=J
250 NEXT J
260 IF FLAG=5 THEN 410
270 IF LEFT$(D$,3)=''?/*'' THEN FLAG=1
280 IF LEFT$(D$,4)=''?/*/'' THEN FLAG=2
290 IF FLAG=3 THEN GOSUB 90:F$=MID$(D$,2,E-2)
300 IF FLAG=1 THEN F$=MID$(D$,4)
310 E=0
320 E=E+1
330 IF A$(E)='' '' AND FLAG=4 AND TRUE=0 THEN PRINT ''FALSO''
340 IF A$(E)='' '' THEN 520
350 IF FLAG=4 AND ''?''+A$(E)+''*''+B$(E)=D$ THEN
    PRINT ''VERDADEIRO'':TRUE=1
360 IF FLAG=3 AND F$=A$(E) THEN PRINT B$(E)
370 IF FLAG=2 THEN PRINT A$(E);''*'';B$(E)
380 IF FLAG=1 AND F$=B$(E) THEN PRINT A$(E)
390 IF E<255 THEN 320
400 REM
410 F$=MID$(D$,4,TRUE-4):G$=MID$(D$,TRUE+7)
```

```
420 E=0
430 E=E+1
440 IF A$(E)='' '' THEN 520
450 IF B$(E) THEN 470
460 IF E<255 THEN 430
470 H=0
480 H=H+1
490 IF B$(H)='' '' THEN 460
500 IF B$(H)=G$ AND A$(E)=A$(H) THEN PRINT A$(E)
510 IF H<255 THEN 480
520 PRINT TAB(5);''>FIM DE RESPOSTA<''
530 GOTO 50
```

12

UM SABOR A PROLOG

Agora que o leitor já tem alguma experiência de trabalho com uma linguagem declarativa, podemos passar à PROLOG. O programa que vou apresentar permitirá ao seu computador executar uma versão limitada do «front-end» da PROLOG intitulado SIMPLE. Chamei-lhe PROLOG-A (que em inglês corresponde a PROLOG-Almost, PROLOG--Quase).

Se bem que o leitor possa aprender bastante sobre a PROLOG lendo este capítulo, e utilizando o programa nele incluído, não se pretende aqui ensinar verdadeiramente a trabalhar nesta linguagem. No entanto, verificará que se adquirir um livro especificamente dedicado à PROLOG ou à micro-PROLOG poderá usar o programa aqui apresentado, juntamente com esse livro, para aprender os aspectos básicos do uso do «front-end» SIMPLE através do qual se usa esta linguagem.

Antes de examinarmos a PROLOG-A, vejamos um exemplo de uso de um verdadeiro compilador da linguagem (tudo o que se apresenta entre /*...*/ é um comentário, semelhante às declarações REM num programa BASIC):

```
/* uma base de dados dia/noite */

mocho é_um animal_nocturno
morcego é_um animal_nocturno
gato é_um animal_diurno
cão é_um animal_diurno

/* as regras */

x dorme_de_dia se x é_um animal_nocturno
x dorme_de_noite se x é_um animal_diurno
x combate y se x é_um animal_diurno
            e y é_um animal_nocturno
```

Como se pode ver, definimos alguns *factos* iniciais, usando a forma é_um(*). Depois indicamos ao computador algumas *regras* que relacionam esses factos entre si. Podemos depois interrogar a máquina do seguinte modo:

```
é(mocho é_um dorme_de_dia)
SIM

que((xy): x combate y)
a resposta é(gato mocho)
a resposta é(gato morcego)
a resposta é(cão mocho)
a resposta é(cão morcego)
```

Vemos aqui que o computador está a usar as regras que lhe fornecemos anteriormente para responder às perguntas. É isto essencialmente o que um sistema pericial faz. A PROLOG pode ser facilmente reconhecida como uma linguagem que parece feita de propósito para a construção de sistemas periciais.

Um programa PROLOG pode também explicar-nos a forma como chegou às suas conclusões.

```
é(gato combate mocho)
SIM

porquê(gato combate mocho)
(gato combate mocho) indicado por
(gato dorme_de_noite) indicado por
(gato é_um) e
(animal_diurno dorme_de_noite)
(mocho dorme_de_dia) indicado por
(mocho é_um animal_nocturno) e
(animal_nocturno dorme_de_dia)
(dorme_de_noite combate dorme_de_dia) dado
```

Vemos portanto que a PROLOG é bastante mais complexa do que a HASTE, se bem que sejam linguagens da mesma família. Espero que as suas experiências com a HASTE o ajudem a compreender o que acontece nos exemplos PROLOG aqui examinados.

Podemos agora observar como trabalha a PROLOG-A.

(*) Ao construir o sistema convém usar a forma «é_um/a», a fim de poder ser usada para termos masculinos e femininos. No texto separam-se os dois casos para maior facilidade de compreensão — *(N. do T.)*

A via marciana

Começámos, tal como no caso da HASTE, por introduzir factos na base de dados. Em PROLOG-A, o pedido de entradas é apresentado pelo símbolo «&.», servindo o ponto para indicar que o computador está a ler o teclado, à espera de uma entrada. Levamos o programa a acrescentar factos à sua base de dados usando um comando muito simples, ADD. Enquanto que em HASTE o material introduzido é formado por duas partes, um sujeito e um predicado (incluindo o verbo), em PROLOG-A esse material é constituído por três partes:

1 — um sujeito (por exemplo JOÃO)
2 — uma relação (como É__PAI__DE)
3 — um outro sujeito ou facto (por exemplo, ROBERTO)

Como podemos ver, as palavras de cada secção estão ligadas por traços. As secções são separadas entre si por espaços. Os espaços e os traços são muito importantes.

Ensinemos algumas coisas ao programa:

```
&.ADD(ZAPPA É_UM MARCIANO)
&.ADD(ZERON É_UM MARCIANO)
&.ADD(EATEE É_UM SUPLEMENTO)
&.ADD(DRINKEE É_UM LÍQUIDO)
&.ADD(LÍQUIDDO É_UMA BEBIDA)
&.ADD(LASER É_UMA ARMA)
&.ADD(YYPRUS É_UM MARCIANO)
```

Podemos descobrir o que o programa sabe usando o comando LIST ALL, que indica simplesmente à PROLOG que deve listar todos os factos que guardou na sua base de dados:

```
&.LIST ALL

ZAPPA É_UM MARCIANO
ZERON É_UM MARCIANO
EATEE É_UM SUPLEMENTO
DRINKEE É_UM LÍQUIDDO
LÍQUIDDO É_UMA BEBIDA
LASER É_UMA ARMA
YYPRUS É_UM MARCIANO
```

Chegamos agora à parte verdadeiramente interessante da PROLOG, onde descobrimos uma capacidade que a HASTE não tem. Em

particular, a PROLOG pode combinar sozinha diferentes entradas de modo a acrescentar informações à sua base de dados. Até agora apenas lhe forneceu os factos, que ela aceitou obedientemente. Ensinar-lhe-emos agora algumas regras que relacionam alguns dos factos em causa.

Vejamos a primeira regra:

```
&.ADD(X É ALIMENTÍCIO SE X É_UMA BEBIDA)
                    REGRA EM COMPILAÇÃO
>LÍQUIDDO É ALIMENTÍCIO
```

Dizemos assim ao computador que um objecto (X é uma variável em PROLOG) é alimentício quando é uma bebida. O programa responde-nos que está a compilar a regra, e imprime tudo o que acrescentou à base de dados em resultado da aplicação em causa. Fornecemos-lhe então uma regra:

```
&.ADD(X É ALIMENTÍCIO SE X É_UM SUPLEMENTO)
                    REGRA EM COMPILAÇÃO
>EATEE É ALIMENTÍCIO
```

Como disse no último parágrafo, X é uma variável em PROLOG. A micro-PROLOG utiliza X, Y, Z, *x*, *y* e *z* como variáveis, mas utilizaremos apenas Z e Y. As variáveis são caixas vazias nas quais podemos pôr o que quisermos. Não podemos usar um nome de variável como nome na base de dados (pelo que não podemos chamar X a nenhum marciano); se o fizermos o programa ficará bastante confuso.

Em seguida pediremos à PROLOG-A que nos diga tudo o que sabe sobre a relação É (que é completamente diferente da relação É__UM):

```
&.LIST É
LIQUIDDO É ALIMENTÍCIO
EATTE É ALIMENTÍCIO
```

E continuamos do mesmo modo, ajudando a PROLOG a formar o seu armazém de factos sobre o universo em que será especialista:

```
&.ADD(X FALA MARCIANÊS SE X É_UM MARCIANO)
                    REGRA EM COMPILAÇÃO
>ZAPPA FALA MARCIANÊS
>ZERON FALA MARCIANÊS
>YYPRUS FALA MARCIANÊS
```

A PROLOG pode também compilar regras relativas a mais do que uma variável, como veremos em seguida:

```
&.ADD(X PROGRAMA Y SE X FALA MARCIANÊS E Y É_UMA ARMA)
                    REGRA EM COMPILAÇÃO
>ZAPA PROGRAMA LASER
>ZERON PROGRAMA LASER
>YYPRUS PROGRAMA LASER
```

O programa ficou assim a conhecer uma relação (PROGRAMA) entre as variáveis X e Y, que existe se X FALA MARCIANÊS E Y É__UMA ARMA.

À medida que avançamos, ensinando gradualmente a PROLOG-A a actuar como sistema pericial no exigente ambiente de Marte, dos seus habitantes e estilo de vida, podemos verificar o que já sabe.

Saberá, por exemplo, que as armas laser são perigosas?

```
&.É(LASER É PERIGOSO)
NÃO
```

Não sabe, pelo que é melhor ensinar-lhe:

```
&.ADD(X É PERIGOSO SE X_É UMA ARMA)
                    REGRA EM COMPILAÇÃO
>LASER É PERIGOSO
```

Verificamos se a lição foi aprendida (usando É para indicar que estamos a fazer uma pergunta):

```
&.É(LASER É PERIGOSO)
SIM
```

Podemos aprender muito interrogando a base de dados. Podemos fazer perguntas em termos de relações:

```
&.LIST FALA
ZAPPA FALA MARCIANÊS
ZERON FALA MARCIANÊS
YYPRUS FALA MARCIANÊS
```

Ou, tal como em HASTE, podemos usar variáveis na pergunta, recorrendo ao uso do comando WHICH:

```
&.WHICH(X:X É_UM MARCIANO)
ZAPPA
```

```
ZERON
YYPRUS
SEM OUTRAS RESPOSTAS
```

Perguntamos aqui qual o objecto sobre o qual se pode dizer que é marciano. Notem-se os espaços na forma interrogativa. Estes são vitais para que a PROLOG-A possa funcionar satisfatoriamente.

Não necessitamos de nos limitar a uma única variável:

```
&.WHICH((X Y) : X É_UM Y)
ZAPPA MARCIANO
ZERON MARCIANO
EATEE SUPLEMENTO
DRINKEE LÍQUIDDO
LÍQUIDO BEBIDA
LASER ARMA
SEM OUTRAS RESPOSTAS
```

Pergunta-se aqui que duas coisas (X e Y) existem relacionadas por É_UM. Note-se que a PROLOG-A imprime os valores de X e Y que encontra sem imprimir a relação. Por outro lado, note-se que (tal como nos exemplos acima) as respostas da PROLOG-A terminam pela indicação SEM OUTRAS RESPOSTAS, notando não existir mais informação sobre o assunto.

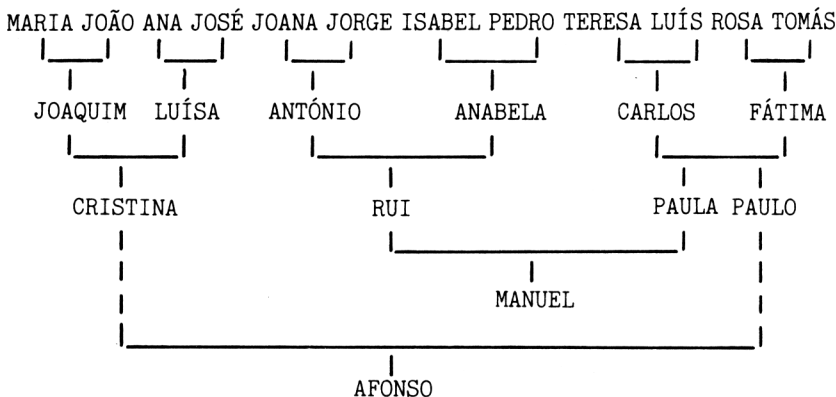
A PROLOG-A possui ainda a capacidade de compilar regras em que existem duas variáveis. É fácil compreender que esta linguagem, apesar de tudo, não pode trabalhar com informação que não possui:

```
&.ADD(X BEBE Y SE X FALA MARCIANÊS E Y É_UMA BEBIDA)
                REGRA EM COMPILAÇÃO
>ZAPPA BEBE LÍQUIDDO
>ZERON BEBE LÍQUIDDO
>YYPRUS BEBE LÍQUIDDO

&.ADD(X FICA_DOENTE_DEPOIS_DE_COMER Y SE X FALA MARCIANÊS
E Y É_UMA CÁPSULA_DE_COMIDA)
                REGRA EM COMPILAÇÃO
DECLARAÇÃO 2 DA ENTRADA NÃO PRESENTE NA BASE DE DADOS

LIST É_UM
ZAPPA É_UM MARCIANO
ZERON É_UM MARCIANO
EATTE É_UM SUPLEMENTO
DRINKEE É_UM LÍQUIDDO
LÍQUIDDO É_UMA BEBIDA
YYPRUS É_UM MARCIANO
```

```
&.ADD(X FICA_DOENTE_DEPOIS_DE_COMER Y SE X FALA MARCIANÊS E Y
É_UM SUPLEMENTO)
                REGRA EM COMPILAÇÃO
>ZAPPA FICA_DOENTE_DEPOIS_DE_COMER EATEE
>ZERON FICA_DOENTE_DEPOIS_DE_COMER EATEE
>YYPRYS FICA_DOENTE_DEPOIS_DE_COMER EATEE
```

Famílias felizes

Por muito instrutiva que esta demonstração seja, o ambiente em Marte é completamente imaginário, e as relações foram inventadas para mostrar como funciona a PROLOG-A. Para mostrarmos que há mais coisas na vida do que vida em Marte, observaremos uma base de dados «verdadeira», muito mais terra-a-terra.

Vamos ensinar ao nosso programa a seguinte árvore genealógica:

Comecemos pela primeira geração:

```
&.ADD(MARIA MÃE_DE JOAQUIM)
&.ADD(JOÃO PAI_DE JOAQUIM)
&.ADD(ANA MÃE_DE LUÍSA)
&.ADD(JOSÉ PAI_DE LUÍSA)
&.ADD(JOANA MÃE_DE ANTÓNIO)
&.ADD(JORGE PAI_DE ANTÓNIO)
&.ADD(ISABEL MÃE_DE ANABELA)
&.ADD(PEDRO PAI_DE ANABELA)
&.ADD(TERESA MÃE_DE CARLOS)
&.ADD(LUÍS PAI_DE CARLOS)
&.ADD(ROSA MÃE_DE FÁTIMA)
&.ADD(TOMÁS PAI_DE FÁTIMA)
```

Verificamos quais os conhecimentos que a PROLOG-A já tem:

```
&.LIST ALL
MARIA MÃE_DE JOAQUIM
JOÃO PAI_DE JOAQUIM
ANA MÃE_DE LUÍSA
JOSÉ PAI_DE LUÍSA
JOANA MÃE_DE ANTÓNIO
JORGE PAI_DE ANTÓNIO
ISABEL MÃE_DE ANABELA
PEDRO PAI_DE ANABELA
TERESA MÃE_DE CARLOS
LUÍS PAI_DE CARLOS
ROSA MÃE_DE FÁTIMA
TOMÁS PAI_DE FÁTIMA
```

Acrescentemos mais alguns ramos familiares:

```
&.ADD(JOAQUIM PAI_DE CRISTINA)
&.ADD(LUÍSA MÃE_DE CRISTINA)
&.ADD(ANTÓNIO PAI_DE RUI)
&.ADD(ANABELA MÃE_DE RUI)
&.ADD(CARLOS PAI_DE PAULA)
&.ADD(FÁTIMA MÃE_DE PAULA)
&.ADD(CARLOS PAI_DE PAULO)
&.ADD(FÁTIMA MÃE_DE PAULO)
&.ADD(RUI PAI_DE MANUEL)
&.ADD(PAULA MÃE_DE MANUEL)
&.ADD(CRISTINA MÃE_DE AFONSO)
&.ADD(PAULO PAI_ AFONSO)
```

No universo marciano, as relações eram do tipo É_UM ou BEBE. Nesta segunda base de dados, temos MÃE_DE e PAI_DE, o que significa que podemos pedir ao programa, por exemplo, que liste todos os MÃE_DE que contém:

```
&.LIST MÃE_DE
MARIA MÃE_DE JOAQUIM
ANA MÃE_DE LUÍSA
JOANA MÃE_DE ANTÓNIO
ISABEL MÃE_DE ANABELA
TERESA MÃE_DE CARLOS
ROSA MÃE_DE FÁTIMA
LUÍSA MÃE_DE CRISTINA
ANABELA MÃE_DE RUI
```

```
FÁTIMA MÃE_DE PAULA
FÁTIMA MÃE_DE PAULO
PAULA MÃE_DE MANUEL
CRISTINA MÃE_DE AFONSO
```

Podemos verificar em seguida os PAI_DE:

```
&.LIST PAI_DE
JOÃO PAI_DE JOAQUIM
JOSÉ PAI_DE LUÍSA
JORGE PAI_DE ANTÓNIO
PEDRO PAI_DE ANABELA
LUÍS PAI_DE CARLOS
TOMÁS PAI_DE FÁTIMA
JOAQUIM PAI_DE CRISTINA
ANTÓNIO PAI_DE RUI
CARLOS PAI_DE PAULA
CARLOS PAI_DE PAULO
RUI PAI_DE MANUEL
PAULO PAI_DE AFONSO
```

A pergunta mais simples que podemos fazer à PROLOG-A respeita à veracidade ou falsidade de uma afirmação:

```
&.É(CRISTINA MÃE_DE MANUEL)
NÃO
&.É(ANTÓNIO PAI_DE RUI)
SIM
&.FÁTIMA MÃE_DE PAULA)
SIM
```

Note-se que, tal como no caso de Marte, as respostas do programa são um SIM ou NÃO. Na primeira pergunta da série anterior foi-nos dito que a mãe de Manuel não é Cristina. É portanto razoável que perguntemos à PROLOG-A quem é a mãe de Cristina:

```
&.WHICH(X : X MÃE_DE CRISTINA)
LUÍSA
SEM OUTRAS RESPOSTAS
```

Esta pergunta, como algumas que já encontrámos em Marte, significa de facto «Qual é o X tal que X é mãe de Cristina?». Os pais podem ser tratados do mesmo modo:

```
&.WHICH(X : X PAI_DE PAULA)
CARLOS
SEM OUTRAS RESPOSTAS
```

Podemos fazer uma pergunta usando duas variáveis, a fim de descobrirmos os nomes de todas as mães e respectivos filhos:

```
&.WHICH (X Y) : X MÃE_DE Y)
MARIA JOAQUIM
ANA LUÍSA
JOANA ANTÓNIO
ISABEL ANABELA
TERESA CARLOS
ROSA FÁTIMA
LUÍSA CRISTINA
ANABELA RUI
FÁTIMA PAULA
FÁTIMA PAULO
PAULA MANUEL
CRISTINA AFONSO
```

Até agora usámos as relações MÃE__DE e PAI__DE. É chegado o momento de ensinarmos alguma coisa à PROLOG-A sobre PROGENITORES (Pai e Mãe):

```
JOÃO PROGENITOR_DE JOAQUIM
JOSÉ PROGENITOR_DE LUÍSA
JORGE PROGENITOR_DE ANTÓNIO
PEDRO PROGENITOR_DE ANABELA
LUÍS PROGENITOR_DE CARLOS
TOMÁS PROGENITOR_DE FÁTIMA
```

Mas além dos pais, também as mães são progenitores. Vamos portanto indicar o facto por uma regra, deixando o resto do trabalho ao computador:

```
&.ADD(X PROGENITOR__DE Y SE X MÃE__DE Y)
                    REGRA EM COMPILAÇÃO
>MARIA PROGENITOR_DE JOAQUIM
                REGRA EM COMPILAÇÃO
                REGRA EM COMPILAÇÃO
>ANA PROGENITOR_DE LUÍSA
                REGRA EM COMPILAÇÃO
                REGRA EM COMPILAÇÃO
>JOANA PROGENITOR_DE ANTÓNIO
                REGRA EM COMPILAÇÃO
                REGRA EM COMPILAÇÃO
>ISABEL PROGENITOR_DE ANABELA
                REGRA EM COMPILAÇÃO
                REGRA EM COMPILAÇÃO
>TERESA PROGENITOR_DE CARLOS
                REGRA EM COMPILAÇÃO
```

Apresentamos depois ao computador uma regra semelhante relativa aos pais, e verificamos se a PROLOG-A acrescentou à sua base de dados todas as consequências destas regras:

```
&.LIST ALL
MARIA MÃE_DE JOAQUIM
JOÃO PAI_DE JOAQUIM
ANA MÃE_DE LUÍSA
JOSÉ PAI_DE LUÍSA
JOANA MÃE_DE ANTÓNIO
JORGE PAI_DE ANTÓNIO
ISABEL MÃE_DE ANABELA
PEDRO PAI_DE ANABELA
TERESA MÃE_DE CARLOS
LUÍS PAI_DE CARLOS
ROSA MÃE_DE FÁTIMA
TOMÁS PAI_DE FÁTIMA
JOAQUIM PAI_DE CRISTINA
LUÍSA MÃE_DE CRISTINA
ANTÓNIO PAI_DE RUI
ANABELA MÃE_DE RUI
CARLOS PAI_DE PAULA
FÁTIMA MÃE_DE PAULA
CARLOS PAI_DE PAULO
FÁTIMA MÃE_DE PAULO
RUI PAI_DE MANUEL
PAULA MÃE_DE MANUEL
CRISTINA MÃE_DE AFONSO
PAULO PAI_DE AFONSO
MARIA PROGENITOR_DE JOAQUIM
ANA PROGENITOR_DE LUÍSA
JOANA PROGENITOR_DE ANTÓNIO
ISABEL PROGENITOR_DE ANABELA
TERESA PROGENITOR_DE CARLOS
ROSA PROGENITOR_DE FÁTIMA
LUÍSA PROGENITOR_DE CRISTINA
ANABELA PROGENITOR_DE PAULA
FÁTIMA PROGENITOR_DE PAULO
PAULA PROGENITOR_DE MANUEL
CRISTINA PROGENITOR_DE AFONSO
JOÃO PROGENITOR_DE JOAQUIM
JOSÉ PROGENITOR_DE LUÍSA
JORGE PROGENITOR_DE ANTÓNIO
PEDRO PROGENITOR_DE CARLOS
TOMÁS PROGENITOR_DE FÁTIMA
```

```
JOAQUIM PROGENITOR_DE CRISTINA
ANTÓNIO PROGENITOR_DE RUI
CARLOS PROGENITOR_DE PAULA
CARLOS PROGENITOR_DE PAULO
RUI PROGENITOR_DE MANUEL
PAULO PROGENITOR_DE AFONSO
```

Podemos indicar ao programa outras relações, especializando o nosso sistema cada vez mais:

```
&.ADD(X FILHO/A_DE Y SE Y PROGENITOR_DE X)
                    REGRA EM COMPILAÇÃO
>JOAQUIM FILHO/A_DE MARIA
>JOAQUIM FILHO/A_DE JOÃO
>LUÍSA FILHO/A_DE JOANA
```

e assim por diante.

Dizemos em seguida à PROLOG-A que MARIA CASOU_COM JOÃO, dando-lhe depois a conhecer o seguinte segredo:

```
&.AD(X EXPOSA_DE Y SE X CASOU_COM Y)
                 REGRA EM COMPILAÇÃO
                 REGRA EM COMPILAÇÃO
                 REGRA EM COMPILAÇÃO
>MARIA ESPOSA_DE JOÃO

&.ADD(X CASOU_COM Y SE X PAI_DE LUÍSA E Y MÃE_DE LUÍSA)
              REGRA EM COMPILAÇÃO
>ANA CASOU_COM JOSÉ
```

O nosso sistema só será adequado se pudermos recuperar informações de base de dados. Podemos fazer perguntas curtas e directas sobre um dado indivíduo:

```
&.É(ISABEL MÃE_DE ANABELA)
SIM
&.WHICH(X : X PAI_DE ANABELA)
PEDRO
SEM OUTRAS RESPOSTAS
```

ou ainda verificar grupos relacionados:

```
&.WHICH((X Y) : X PAI_DE Y)
JOÃO JOAQUIM
JOSÉ LUÍSA
```

```
JORGE ANTÓNIO
PEDRO ANABELA
LUÍS CARLOS
TOMÁS FÁTIMA
JOAQUIM CRISTINA
ANTÓNIO RUI
CARLOS PAULA
CARLOS PAULO
RUI MANUEL
PAULO AFONSO

SEM OUTRAS RESPOSTAS

&.WHICH((X Y) : X FILHO/A_DE Y)
JOAQUIM MARIA
JOAQUIM JOÃO
LUÍSA ANA
LUÍSA JOSÉ
ANTÓNIO JOANA
ANTÓNIO JORGE
ANABELA ISABEL
ANABELA PEDRO
CARLOS TERESA
CARLOS LUÍS
```

e assim por diante.

O jogo dos números

A PROLOG-A tem algumas capacidades no campo numérico, como veremos em seguida:

```
&.É(SOMA (12 13 25))
SIM
```

Perguntámos aqui ao programa se a soma 12 e 13 é 25. Vejamos três outros exemplos:

```
&.É(SOMA(12 14 20))
NÃO
&.É(SOMA(5.4 -8 -3.6))
NÃO
&.É(SOMA(-7 -9 -16))
SIM
```

Este uso é ainda bastante limitado. Mas podemos usar SOMA para determinar valores desconhecidos. Primeiramente vamos somar dois números:

```
&.WHICH(X : SOMA(5.4 -8 X))
-2.6
SEM OUTRAS RESPOSTAS
```

Vemos aqui a já familiar forma «Qual é o X tal que X é igual a 12 mais 32». Alternativamente, podemos usar a forma «Qual é o X que, quando somado a um número conhecido, produz um resultado conhecido?»:

```
&.WHICH(X : SOMA(X 45 87))
42
SEM OUTRAS RESPOSTAS
```

Se colocarmos o valor desconhecido noutra posição, a PROLOG-A fará uma subtracção:

```
&.WHICH(X : SOMA(123 X -98)
-221
SEM OUTRAS RESPOSTAS
```

Perguntamos assim «Qual é o X que resulta da subtracção do segundo número pelo terceiro?».

A PROLOG-A pode igualmente verificar se um número é ou não inteiro:

```
&.É(37 INT)
SIM
&.É(-987 INT)
SIM
&.É(88.7 INT)
NÃO
&.É(8870 INT)
SIM
```

Podemos pedir-lhe que indique a porção inteira de um número representado em vírgula flutuante:

```
&.WHICH(X : 1234.8 INT X)
1234
&.WHICH(X : -8889.2987 INT X)
-8890
```

Usa-se VEZES («TIMES») de uma forma semelhante a SOMA («SUM»):

```
&.É(VEZES(9 3 27))
SIM
&.É(VEZES(27 3 99))
NÃO
&.WHICH(X : VEZES(12 76 X))
912
SEM OUTRAS RESPOSTAS
```

Tal como SOMA pode ser usada para subtrair, também VEZES pode ser usado para fazer divisões. Colocando X na primeira ou na segunda posições obtemos a mesma resposta:

```
&.WHICH(X : VEZES(12 X 76))
6.3333333333333
SEM OUTRAS RESPOSTAS
&.WHICH(X : VEZES(X 12 76))
6.3333333333333
SEM OUTRAS RESPOSTAS
```

Neste caso, a PROLOG-A responde à pergunta «Qual é o X tal que X é igual a 76 dividido por 12?».

O programa pode comparar o valor de dois números (e também a posição relativa de palavras, sendo as que ocorrem primeiro em ordem alfabética consideradas como «menos do que» as que ocorrem depois):

```
&.É(123 MENOS_DE 97)
NÃO
&.É(97 MENOS_DE 123)
SIM
&.É(TIM MENOS_DE HARTNELL)
NÃO
&.É(HARTNELL MENOS_DE COMPUTADOR)
NÃO
&.É(A MENOS_DE B)
SIM
&.É(AAAA1 MENOS_DE A2)
NÃO
&.É(AA MENOS_DE A1)
NÃO
&.É(CURTO MENOS_DE CURTOS)
SIM
```

Empregamos aqui MENOS__DE em vez do operador de facto usado pela linguagem, LESS, sendo este o que o leitor deverá usar se experimentar estes exemplos depois de introduzir a listagem a PROLOG-A no seu computador. O mesmo se passa em relação a SOMA e a VEZES, que deverão ser expressos por SUM e TIMES; e a operadores como SE ou E, indicados por IF ou AND.

A partir do que se disse será fácil ao leitor concluir que o trabalho em matemáticas usando a PROLOG-A não é uma tarefa particularmente fácil. Tanto a PROLOG como a LISP não foram de facto concebidas para trabalho matemático, mas tentou-se em ambas incluir alguns operadores que permitissem esse trabalho; a sintaxe deste segue a sintaxe de «processamento de listas» habitual nestas linguagens.

13

A LISTAGEM PROLOG-A

A PROLOG-A é de facto uma simulação de PROLOG em BASIC, e não um verdadeiro interpretador daquela linguagem; não se comporta portanto em todas as situações da mesma maneira que o «front-end» SIMPLE da micro-PROLOG. No entanto, é suficientemente sólida na maior parte das situações (se bem que a compilação das regras não seja tão directa como seria de desejar). O programa será porém suficiente para dar ao leitor um instrumento que o ajudará a aprender PROLOG, e com o qual poderá fazer muitas experiências.

A listagem que se segue é bastante extensa. Para a tornar mais fácil de introduzir na máquina, escrevi deliberadamente a PROLOG-A de modo a permitir-lhe deixar de lado toda a parte matemática sem afectar o funcionamento do resto (pelo que o leitor poderá acrescentar a parte matemática mais tarde, depois de experimentar um pouco a listagem mais reduzida). Para eliminar a secção matemática, basta não introduzir as linhas 2430 a 3260. Poderá também (mas não o aconselho a fazê-lo) omitir as linhas 220, 230, 240 e 250. Se o fizer, terá de as escrever novamente quando acrescentar as matemáticas.

Como se indica na linha 20, todas as entradas devem ser feitas em maiúsculas. O programa pode tratar até 1000 linhas na sua base de dados, podendo compilar até 100 factos a partir de uma regra (se bem que seja improvável que tal seja necessário). No entanto, o programa funciona de forma cada vez mais lenta (como seria de esperar) à medida que a quantidade de informação cresce. Isto não é grave, dado que a resposta à maior parte das perguntas é dada em segundos.

Uma vez mais se recorda que os operadores são usados pelo programa na sua forma inglesa, como se indicou no final do capítulo anterior.

```
10 REM PROLOG-A (SIMPLE)
20 REM TODAS AS ENTRADAS DEVEM SER FEITAS EM MAIÚSCULAS
30 GOTO 50
40 PRINT ''SEM OUTRAS RESPOSTAS'':RETURN
50 GOSUB 3270:REM INICIALIZAR
60 REM **********
```

```
70 PRINT
80 INPUT ''&.'',J$
90 IF J$='' '' THEN END
100 IF J$=''LIST ALL'' THEN GOSUB 1860:GOTO 70
110 IF LEFT$(J$,5)=''LIST'' THEN J=MID$(J$,5)+'' '':GOSUB 990:
    GOTO 70
120 IF RIGHT$(J$,1)<>'')'' THEN PRINT ''1.'':INPUT M$:J$=J$+
    M$: GOTO 120
130 LJ=LEN(J$)
140 J$=LEFT$(J$,LJ-1)+'' '':REM tirar ) final, substituir por
    espaço
150 LJ=LEN(J$)
160 FLAG=0
170 IF LEFT$(J$,4)=''ADD(''THEN J$=MID$(J$,5):FLAG=1
180 RULEFLAG=0:PLUSFLAG=0:ARITHFLAG=0
190 FOR R=1 TO LEN(J$)
200 IF MID$(J$,R,4)=''IF'' THEN RULEFLAG=R:FLAG=6
210 IF MID$(J$,R,5)=''AND'' THEN PLUSFLAG=R
220 IF MID$(J$,R,4)=''SUM('' THEN ARITHFLAG=1
230 IF MID$(J$,R,6)=''TIMES('' THEN ARITHFLAG=2
240 IF MID$(J$,R,6)=''LESS'' THEN ARITHFLAG=3
250 IF MID$(J$,R,3)=''INT'' THEN ARITHFLAG=4
260 NEXT R
270 IF LEFT$(J$,3)=''IS('' THEN J$=MID$(J$,4):FLAG=2
280 IF LEFT$(J$,10)=''WHICH(X:''THEN J$=MID$
    (J$,11):FLAG=3
290 IF LEFT$(J$,16)=WHICH((X Y):X'' THEN J$=MID(J$,17):FLAG=4
300 IF FLAG=0 THEN PRINT ''ERRO SINTÁTICO»:GOTO 70
310 LJ=LEN(J$)
320 REM ENVIAR PARA SUBROTINAS RELEVANTES
330 IF PLUSFLAG<>0 THEN GOSUB 1950:GOTO 70:
    REM CODIFICAR REGRA CONTENDO AND
340 IF RULEFLAG<>0 AND FLAG<>5 THEN GOSUB 1110:
    REM CODIFICAR REGRA
350 IF ARITHFMAG<>0 THEN GOSUB 2430:GOTO 70:REM ARITMÉTICA
360 IF RIGHT$(J$,3)=''X'' OR RIGHT$(J$,3)=''Y''
    THEN J$=LEFT$(J$,LJ-2)+'' ''
370 LJ=LEN(J$)
380 IF FLAG=1 THEN GOSUB 440:REM ADD
390 IF FLAG=2 THEN GOSUB 520:REM IS
400 IF FLAG=3 THEN GOSUB 610:REM WHICH
410 IF FLAG=4 THEN GOSUB 830:REM WHICH2
420 GOTO 70
430 REM *******
440 REM ADD
```

```
450 K=0
460 K=K+1
470 IF Z$(K)='' ''THEN Z$(K)=J$:RETURN
480 IF K<1000 THEN 460
490 PRINT ''MEMÓRIA PREENCHIDA''
500 RETURN
510 REM ***********
520 REM IS
530 K=0
540 K=K+1
550 IF Z$(K)='' ''THEN 580
560 IF Z$(K)=J$ THEN PRINT ''SIM'':GOTO 590
570 IF K<1000 THEN 540
580 PRINT ''NÃO''
590 RETURN
600 REM **********
610 REM WHICH
620 IF LEFT$(J$,1)=''X'' THEN 710
630 J$=LEFT$(J$,LJ-1)
640 K=0
650 K=K+1
660 IF Z$(K)='' ''THEN 690
670 IF J$=LEFT$(Z$(K),LEN(J$)) THEN PRINT RIGHT$
    (Z$(K),(LEN(Z$(K))-LEN(J$)))
680 IF K<1000 THEN 650
690 GOSUB 40
700 RETURN
710 REM PERGUNTAS COMEÇADAS POR X
720 J$=MID$(J$,3,LEN(J$)-3)
739 LJ=LEN(J$)
749 K=0
750 K=K+1
760 IF Z$(K)='' ''THEN 800
770 Q$=MID$(Z$(K),LEN(Z$(K))-LJ,LJ)
780 IF Q$=J$ THEN PRINT LEFT$(Z$(K),LEN(Z$(K))-(LJ+2))
790 IF K<1000 THEN 750
800 GOSUB 40
810 RETURN
820 REM **********
830 REM WHICH2
840 J$=LEFT$(J$,LJ-2)
850 LJ=LEN(J$)
860 K=0
870 K=K+1
880 IF Z$(K)='' ''THEN 960
```

```
890 LFLAG=0
900 FOR L=1 TO LEN(Z$(K))-LJ
910 IF MID$(Z$(K),L,LJ)=J$ THEN LFALG=L
920 NEXT L
930 IF FLAG=0 THEN 950
940 PRINT LEFT$(Z$(K), LFLAG-2);MID$(Z$(K), (LFLAG+LJ))
950 IF K<1000 THEN 870
960 GOSUB 40
970 RETURN
980 REM **********
990 REM LIST
1000 K=0
1010 K=K+1
1020 IF Z$(K)='' '' THEN RETURN
1030 LFLAG=0
1040 FOR L=1 TO LEN(Z$(K))-LEN(J$)
1050 IF MID$(Z$(K),L,LEN(J$))=J$ THEN LFLAG=1
1060 NEXT L
1070 IF LFLAG=1 THEN PRINT Z$(K)
1080 IF K<1000 THEN 1010
1090 RETURN
1100 REM **********
1110 REM FORMAR REGRAS
1120 R=RULEFLAG
1130 E$=LEFT$(J$,R):F$=MID$(J$,R+4)
1140 IF LEFT$(E$,1)<>''X'' THEN PRINT ''ERRO DE REGRA'':GOTO 70
1150 REM A LINHA SEGUINTE DETECTA ENTRADAS COMO X COME Y SE X É-UM Y
1160 IF RIGHT$(F$,2)=''Y'' THEN 1390
1170 PRINT TAB(18);''REGRA EM COMPILAÇÃO''
1180 FOR T=1 TO 100
1190 R$(T)='' ''
1200 NEXT T
1210 E$=MID$(E$,3):F$=MID$(F$,3)
1220 K=0:RR=0
1230 K=K+1
1240 IF Z$(K)='' '' THEN 1300
1250 IF RIGHT$(Z$(K),LEN(F$))<>F$ THEN 1370
1260 RR=RR+1
1270 R$(RR)=LEFT$(Z$(K),LEN(Z$(K))-LEN(F$)))+E$
1280 PRINT''>'':R$(RR)
1290 GOTO 1230
1300 IF RR=0 THEN RETURN
1310 RC=0
1320 RC=RC+1
1330 Z$(K)=R$(RC)
```

```
1340 IF K<1000 THEN K=K+1
1350 IF RC<RR THEN 1320
1360 RETURN
1370 IF K<1000 THEN 1230
1380 RETURN
1390 REM REGRA COM DUAS VARIÁVEIS
1400 FOR T=1 TO 100
1410 R$(T)='' ''
1420 NEXT T
1430 K=0:RR=0
1440 IF K=1000 THEN RETURN
1450 K=K+1
1460 IF Z$(K)='' ''THEN 1770
1470 REM SEPARAR EM TRÊS PALAVRAS
1480 Q$=Z$(K)
1490 J=0
1500 J=J+1
1510 IF MID$(Q$,J,1)='' ''THEN 1540
1520 IF J<LEN(Q$) THEN 1500
1530 PRINT ''ERRO DE COMPILAÇÃO DA REGRA'':GOTO 70
1540 A$=LEFT$(Q$,J)
1550 Q$=MID$(Q$,J+1)
1560 J=0
1570 J=J+1
1580 IF MID$(Q$,J,1)='' ''THEN 1610
1590 IF J<LEN(Q$) THEN 1570
1600 PRINT ''ERRO DE COMPILAÇÃO DA REGRA'':GOTO 70
1610 B$=LEFT$(Q$,J)
1620 Q$=MID$(Q$,J+1)
1630 J=0
1640 J=J+1
1650 IF MID$(Q$,J,1)='''THEN 1680
1660 IF J<LEN(Q$) THEN 1640
1670 PRINT ''ERRO DE COMPILAÇÃO'':GOTO 70
1680 PRINT TAB(18);''REGRA EM COMPILAÇÃO''
1690 C$=LEFT$(Q$,J)
1700 M$=MID$(F$,3,LEN(B$))
1710 IF B$<>M$ THEN 1440
1720 RR=RR+1
1730 N$=MID$(E$,3,LEN(E$)-4)
1740 R$(RR)=A$+N$+C$
1750 PRINT ''>'';R$(RR)
1760 GOTO 1440
1770 IF RR=0 THEN RETURN
1780 M=0
```

```
1790 M=M+1
1800 IF M>RR THEN RETURN
1810 Z$(K)=R$(M)
1820 IF K=1000 THEN PRINT ''MEMÓRIA PREENCHIDA'': GOTO 70
1830 K=K+1
1840 GOTO 1790
1850 REM **********
1860 REM LIST ALL
1870 PRINT
1880 K=0
1890 K=K+1
1900 IF Z$(K)='' ''THEN RETURN
1910 PRINT Z$(K)
1920 IF K<1000 THEN 1890
1930 RETURN
1940 REM **********
1950 REM FORMAR REGRAS COM 'AND' DO SEGUINTE TIPO:
1960 REM (X COME Y SE X É_UM PÁSSARO E Y É_COMPRADO_EM CAIXAS)
1970 REM A DECLARAÇÃO X DEVE ESTAR NA LISTA ANTES DE Y PARA QUE
     TODOS OS EXEMPLOS SEJAM CODIFICADOS
1980 REM SEPARAÇÃO EM SECÇÕES
1990 J$=MID(J$,2):REM RETIRAR ''X''
2000 PRINT TAB(20):''REGRA EM COMPILAÇÃO''
2010 J=1
2020 J=J+1
2030 IF MID$(J$,J,1)='' ''THEN 2060
2040 IF J<LEN(J$) THEN 2020
2050 PRINT ''ERRO DE COMPILAÇÃO DA REGRA'':RETURN
2060 A$=LEFT$(J$,J):REM RELAÇÃO 1
2070 J$=MID$(J$,J+7):REM SEPARAR SEGUNDA RELAÇÃO
2080 J=1:COUNT=0
2090 J=J+1
2100 IF MID$(J$,J,1)='' ''THEN COUNT=COUNT+1
2110 IF COUNT=2 THEN 2140
2120 IF J<LEN(J$) THEN 2090
2130 PRINT ''ERRO DE COMPILAÇÃO DA REGRA'':RETURN
2140 B$=LEFT$(J$,J):REM DECLARAÇÃO1
2150 C$=MID$(J$,J+6)$REM DECLARAÇÃO2
2160 IF C$='' ''THEN PRINT ''ERRO DE COMPILAÇÃO DA REGRA'':
     RETURN
2170 REM PERCORRER BASE DE DADOS
2180 FOR T=1 TO 200
2190 R$(T)='' ''
2200 NEXT T
2210 R1=0:R2=99
```

```
2220 K=0
2230 K=K+1
2240 IF Z$(K)='' ''THEN 2310
2250 IF R1=99 OR R2=200 THEN PRINT ''MEMÓRIA INSUFICIENTE'':
     GOTO 2310
2260 LB=LEN(B$)
2270 IF RIGHT$(Z$(K),LB)=B$ THEN R1=R1+1:R$(R1)=LEFT$(Z$(K),
     LEN(Z$(K))-LB)
2280 LC=LEN(C$)
2290 IF RIGHT$(Z$(K),LC)=C$ THEN R2=R2+1:R$(R2)=LEFT$(Z$(K),
     LEN(Z$(K))-LC)
2300 IF K<1000 THEN 2230
2310 IF R$(100)='' ''THEN PRINT ''A SEGUNDA DECLARAÇÃO NÃO SE
     ENCONTRA NA BASE DE DADOS'':RETURN
2320 REM CODIFICAR REGRAS
2330 R1=0:R2=99
2340 R2=R2+1
2350 R1=R1+1
2360 IF R$(R1)>'' ''AND R$(R2)>'' ''THEN Z$(K)=R$(R1)+A$+
     R$(R2)+ '' ''
2370 PRINT''>'';Z$(K)
2380 K=K+1
2390 IF R$(R2+1)<>'' ''THEN 2340
2400 IF R$(R1+1)<>'' ''THEN 2350
2410 RETURN
2420 REM **********
2430 REM ARITMÉTICA
2440 LJ=LEN(J$)
2450 IF ARITHFLAG<3 THEN GOSUB 2490
2460 IF ARITHFLAG=3 THEN GOSUB 2890
2470 IF ARITHFLAG=4 THEN GOSUB 3080
2480 RETURN
2490 REM SUM, TIMES
2500 J$=MID$(J$,5,LJ-5)
2510 IF LEFT$(J$,2)=''S(''THEN J$=MID$(J$,3)
2520 LJ=LEN(J$)
2530 K=0
2540 K=K+1
2550 IF MID$(J$,K,1)='' ''THEN A$=LEFT$(J$,K-1):J$=MID$
     (J$,K+1):GOTO 2580
2560 IF K<LJ THEN 2540
2570 PRINT TAB(12);''ERRO ARITMÉTICO'':RETURN
2580 LJ=LEN(J$)
2590 K=0
2600 K=K+1
```

```
2610  ID MID$(J$,K,1)='' ''THEN B$=LEFT$(J$,K-1):J$=MID$
     (J$,K+1):GOTO 2640
2620 IF K<LJ THEN 2600
2630 PRINT TAB(12);''ERRO ARITMÉTICO'':RETURN
2640 LJ=LEN(J$)
2650 K=0
2660 K=K+1
2670 IF MID$(J$,K,1)='')'' THEN C$=LEFT$(J$,K-1):GOTO 2700
2680 IF K<LJ THEN 2660
2690 PRINT TAB(12);''ERRO (DEMASIADAS VARIÁVEIS)'': RETURN
2700 AN=0:BN=0:CN=0
2710 IF ASC(A$)>58 THEN AN=1
2720 IF ASC(B$)>58 THEN BN=2
2730 IF ASC(C$)>58 THEN CN=4
2740 GUIDE=AN+BN+CN:IF GUIDE=3 OR GUIDE=5 OR GUIDE=6 THEN 2690
2750 IF ARITHFLAG=2 THEN 2820:REM TIMES
2760 IF GUIDE>0 THEN 2790
2770 IF VAL(A$)+VAL(B$)=VAL(C$) THEN PRINT ''SIM'':RETURN
2780 PRINT ''NÃO'':RETURN
2790 IF GUIDE=1 THEN PRINT VAL(C$)-VAL(B$):GOSUB40:RETURN
2800 IF GUIDE=2 THEN PRINT VAL(C$)-VAL(A$):GOSUB 40:RETURN
2810 PRINT VAL(A$)+VAL(B$):GOSUB 40:RETURN
2820 REM TIMES
2830 IF GUIDE>0 THEN 2860
2840 IF VAL(A$)*VAL(B$)=VAL(C$) THEN PRINT ''SIM'':RETURN
2850 PRINT ''NÃO'':RETURN
2860 IF GUIDE=1 THEN PRINT VAL(C$)/VAL(B$):GOSUB 40:RETURN
2870 IF GUIDE=2 THEN PRINT VAL(C$)/VAL(A$):GOSUB 40:RETURN
2880 PRINT VAL(A$)*VAL(B$):GOSUB 40:RETURN
2890 REM **** MENOS ****
2900 NF=0
2910 IF ASC(J$)<58 THEN NF=1:REM NÚMEROS
2920 COUNT=0
2930 K=0
2940 K=K+1
2950 IF MID$(J$,K,1)='' ''THEN COUNT=COUNT+1
2960 IF COUNT=2 THEN 3000
2970 IF K<LEN(J$) THEN 2940
2980 PRINT TAB(20);''ERRO DE COMPARAÇÃO''
2990 RETURN
3000 B$=MID$(J$,K+1)
3010 A$=LEFT$(J$,K-6)
3020 IF NF=1 THEN 3050
3030 IF A$<B$ THEN PRINT ''SIM'':RETURN
3040 PRINT ''NÃO'':RETURN
```

```
3050 REM*NÚMEROS*
3060 IF VAL(A$)<VAL(B$) THEN PRINT ''SIM'':RETURN
3070 PRINT ''NÃO'':RETURN
3080 REM ***** INT ****
3090 IF RIGHT$(J$,2)=''X'' THEN 3190
3100 K=0
3110 K=K+1
3120 IF MID$(J$,K,1)='' ''THEN 3160
3130 IF K<LEN(J$) THEN 3110
3140 PRINT TAB(20);''ERRO DE ARITMÉTICA''
3150 RETURN
3160 A=VAL(LEFT$(J$,K-1))
3170 IF INT(A)=A THEN PRINT ''SIM'':RETURN
3180 PRINT ''NÃO'':RETURN
3190 K=0
3200 K=K+1
3210 IF MID$(J$,K,1)='' ''THEN 3240
3220 IF K<LEN(J$) THEN 3200
3230 PRINT TAB(20);'' ERRO ARITMÉTICO'':RETURN
3240 PRINT INT(VAL(LEFT$(J$,K-1)))
3250 RETURN
3260 REM**********
3270 REM INICIALIZAR
3280 CLS
3290 DIM Z$(1000),R$(200)
3300 RETURN
```

14

UMA VERSÃO DA LISP

A PROLOG, que estudámos nos capítulos anteriores, é um desenvolvimento da LISP (tal como as linguagens Smalltalk e Logo). A LISP é muito mais difícil de aprender a usar do que o «front-end» SIMPLE da PROLOG, mas, juntamente com esta maior dificuldade de aprendizagem, permite uma maior flexibilidade e potência. Depois de dominar a PROLOG-A, o leitor estará em posição de mergulhar na LISP.

As palavras

Existem apenas dois tipos de palavras em LISP, os átomos e as listas. Um átomo é uma palavra (ou uma combinação de letras e números, começando por uma letra). Não pode conter espaços nem quaisquer caracteres que não sejam letras ou números. Vejamos exemplos destes átomos.

COMPUTADOR BOMBA R2D2 C3P0

Uma lista é formada por átomos e outras listas. Começa por um parêntesis esquerdo, seguido de átomos e listas, e termina por um parêntesis direito. Podem existir outros pares de parêntesis no interior do par principal. A lista mais simples conterá um único átomo, por exemplo (COMPUTADOR). Pode conter mais de um átomo, cada um deles separado dos outros por espaços: (COMPUTADOR BOMBA R2D2). A LISP permite o uso de listas vazias, indicadas por (). Chama-se-lhes listas vazias ou nulas.

Uma lista pode conter outras listas. Pensemos na lista (COMO ESTÁ VOCÊ). Poderia encontrar-se no interior de outra lista: (A PERGUNTA É (COMO ESTÁ VOCÊ)). As listas (e a proliferação de parêntesis) podem tornar-se bastante complexas, como acontece na lista ((COMO (PODERÁS TU (FAZÊ-LO)) AGORA) JOÃO). De facto o grande número de parêntesis deu origem a uma teoria segundo a qual a palavra LISP, em vez de significar LISt Processing (Processamento de Listas), significaria de facto «Lots of Infuriatingly Stupid Parentheses» (Muitos Parêntesis Enervantemente Estúpidos).

Não se preocupe se o assunto lhe parece já complicado. Os nossos programas de emulação da LISP, EASLE e SSLISP, conduzi-lo-ão sem grande esforço no aprendizado da linguagem, fazendo coisas como a contagem dos parêntesis, garantindo que não ocorrem erros.

As funções básicas

Existem seis funções básicas em LISP. O nosso primeiro programa, EASLE (Easy And Simple Lisp-like Exerciser), permitir-lhe-á experimentar estas funções. As seis funções são CAR, CDR, CONS, ATOM, NULL e EQ. Num verdadeiro sistema LISP, podem-se usar apenas estas seis funções para criar praticamente qualquer função de computador. O leitor compreenderá dentro em pouco o que fazem as funções em causa, ao observar o seu uso em EASLE. Tanto esta como a SSLISP seguem a sintaxe EVQ-LISP, que é uma das mais fáceis de usar. Os outros dialectos, incluindo EV-LISP ou P-LISP, exigem uma sintaxe de entrada ligeiramente diferente. No entanto, todas as LISP's, especialmente ao nível mais básico, são bastante semelhantes, e o hábito que o leitor pode ganhar trabalhando com os meus programas facilitar-lhe-á a vida ao tentar estudar qualquer dos outros sistemas.

CAR

Esta função é usada para obter o primeiro elemento de uma lista. É usada com um par de parêntesis próprio, encerrando a lista que se pretende verificar (que se encontra no interior de um outro par de parêntesis). O caso mais simples é:

```
>CAR ((ISTO É FÁCIL))
  VALUE IS...
  ISTO
```

No caso seguinte, o primeiro elemento da lista é uma outra lista (ISTO É), como se pode verificar:

```
>CAR (((ISTO É) FÁCIL))
  VALUE IS...
  (ISTO É)
```

Nos casos em que a PROLOG responde SIM ou NÃO, a LISP responde um T (indicando «True» ou verdadeiro) ou um NIL (indicando falso). Uma lista vazia () é referida como NIL pelo sistema, como se

vê aqui (a lista vazia é o primeiro elemento da lista a que aplicamos a função CAR):

```
>CAR ((() (ISTO É FÁCIL)))
   VALUE IS...
   NIL
```

O leitor recorda certamente que já foi dito que EASLE e SSLISP contam os parêntesis das entradas. Poderá ver como isto acontece aqui, dado que esqueci acidentalmente o último:

```
>CAR ((ISTO É) (A MADRUGADA)
 ->MISSING  )
  >  )
   VALUE IS...
   (ISTO É)
>
```

CDR

Pronunciando-se «cooder», esta função responde a lista que resta depois de eliminado o CAR, como se pode verificar nos exemplos seguintes:

```
>CDR ((ERA DE AQUÁRIO))
   VALUE IS...
   (DE AQUÁRIO)

>CDR (((A ERA DE) AQUÁRIO))
   VALUE IS...
   (AQUÁRIO)

>CDR (((PRINCÍPIO DA ERA)))
   VALUE IS...
   NIL

>CDR ((()(ERA DO SPECTRUM)))
   VALUE IS...
   ((ERA DO SPECTRUM))
```

CONS

Esta função junta listas (ao contrário de CAR e CDR que as dividem). Combina dois argumentos, o segundo dos quais deve ser uma

lista, como deve ser claro a partir do exemplo que se segue, uma execução do programa EASLE:

```
>CONS (OS COMPUTADORES (FAZEM POUCO))
  VALUE IS...
  (OS COMPUTADORES FAZEM POUCO)

>CONS ((OS COMPUTADORES) (TRABALHAM DEVAGAR))
  VALUE IS...
  ((OS COMPUTADORES) TRABALHAM DEVAGAR)

>CONS ((A ERA) ((ANTES (DA BELEZA))))
  VALUE IS...
  (((A ERA) (ANTES (DA BELEZA)))

>CONS (AQUÁRIO ())
  VALUE IS...
  (AQUÁRIO)
```

Pode-se ver que uma lista a que se aplica a função CONS é constituída por uma CAR e uma CDR, os dois argumentos da lista.

ATOM

Depois de CAR, CDR e CONS passamos a ATOM, muito mais simples que as três precedentes. O leitor recorda certamente que a LISP responde T ou NIL a uma pergunta do utilizador. ATOM (juntamente com as duas funções que observaremos em seguida, EQ e NULL) é uma função de *teste,* e produz precisamente uma destas duas respostas. ATOM responde T se o único elemento da lista que está a ser considerada é um átomo. O funcionamento do programa deve tornar isto claro:

```
>ATOM (SPITFIRE)
  VALUE IS...
  T

>ATOM ((ERA DE AQUÁRIO))
  VALUE IS...
  NIL

>ATOM ((AQUÁRIO))
  VALUE IS...
  NIL
```

EQ

Vimos atrás que ATOM produz um T se o argumento único é um átomo, e NIL no caso contrário. EQ (de «igual») produz T se os dois argumentos de EQ são átomos idênticos; senão produz um NIL.

Vejamos o funcionamento de EQ:

```
>EQ (AQUÁRIO AQUARIUS)
  VALUE IS...
  NIL

>EQ (AQUÁRIO AQUÁRIO)
  VALUE IS...
  T
```

NULL

Finalmente, estudaremos a sexta função, designada por NULL. Esta possui um único argumento, e produz T no caso de o agrupamento ser a lista vazia (), e NIL se não o for:

```
>NULL (())
  VALUE IS...
  T

>NULL ((()))
  VALUE IS...
  NIL

>NULL ()
  ILLEGAL - NULL NEEDS ARGUMENT
  VALUE IS...

>NULL ((PRINCÍPIO DA ERA))
  VALUE IS...
  NIL
```

EASLE

A LISP (ao contrário da EASLE) permite-nos escrever expressões sofisticadas que o computador avaliará sem dificuldades, como:

```
CONS (CAR ((SALSICHAS)) (CONS (FRITAS)
           CDR ((NUNCA SABEM BEM))))
```

Isto produz:

```
VALUE IS...
(SALSICHAS FRITAS SABEM BEM)
```

A SSLISP (que significa «Single Statement LISP) também não permite o uso de expressões múltiplas, o que limita seriamente o seu uso. No entanto, se bem que a EASLE e a SSLISP não possam ser usadas para criar sistemas periciais, ambas podem certamente ser usadas para dar ao leitor os conceitos básicos necessários para usar uma LISP mais completa.

Introduza no seu computador a listagem da EASLE apresentada a seguir, e use-a para avaliar diversas listas por si criadas:

```
10 REM EASLE
20 CLS
30 DIM X(40),Y(40),Z(40)
40 PRINT
50 INPUT''>'',A$
60 IF A$='' ''THEN END
70 REM **********
80 FOR J=1 TO 40
90 X(J)=0:Y(J)=0:Z(J)=0
100 NEXT J
110 REM **********
120 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
130 FOR J=1 TO LEN(A$)
140 B$=MID$(A$,J,1)
150 IF B$=''(''THEN S=S+1:Z(S)=J:IF T=0 THEN CFIRST=J
160 IF B$='')'' THEN T=T+1:Y(T)=J:IF CSECND<>0 AND EDGE=0
    THEN EDGE=J-1
170 IF T=1 AND B$='')'' THEN CSECND=J
180 IF B$'' '' THEN R=R+1:X(R)=J
190 NEXT J
200 IF S=T THEN 260:REM EQUILÍBRIO DE ()
210 IF S<T THEN PRINT '' -> MISSING (''
220 IF S>T THEN PRINT '' ->MISSING )''
230 INPUT '' > '',B$
240 A$=A$+B$
250 GOTO 80
260 FLAG=0
270 IF LEFT$(A$,5)=''CAR (''THEN FLAG=1
280 IF LEFT$(A$,5)=''CDR (''THEN FLAG=2
290 IF LEFT$(A$,6)=''CONS (''THEN FLAG=3
300 IF LEFT$(A$,6)=''ATOM (''THEN FLAG=4
```

```
310 IF LEFT$(A$,4)=''EQ (''THEN FLAG=5
320 IF LEFT$(A$,6)=''NULL (''THEN FLAG=6
330 ON FLAG GOSUB 420, 470, 550, 690, 780, 920
340 IF FLAG<>0 THEN GOSUB 360
350 GOTO 40
360 REM ** RESPOSTA **
370 PRINT '' VALUE IS...''
380 IF B$<>''()'' THEN PRINT'' '';B$
390 IF B$=''()'' THEN PRINT'' NIL''
400 RETURN
410 REM **********
420 REM ** CAR **
430 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z(2))
440 IF SD>2 THEN B$=MID$(A$,CFIRST,CSECND-CFIRST+1)
450 RETURN
460 REM **********
470 REM ** CDR **
480 GOSUB 420
490 LB=LEN(B$)+7
500 B$=''(''+MID$(A$,LB,EDGE-1)
510 IF RIGHT$(B$,2)=''))'' THEN B$=LEFT$(B$,LEN(B$)-1)
520 IF MID$(B$,2,1)='' '' THEN B$''(''+MID$(B$,3)
530 RETURN
540 REM **********
550 REM **CONS **
560 B$=MID$(A$,7,LEN(A$)-1)
570 J=0
580 IF LEFT$(B$,1)=''(''THEN J=1
590 J=J+1
600 IF MID$(B$,J,1)=''(''THEN 630
610 IF J<LEN(B$) THEN 590
620 B$='' >CONS ERROR<'':RETURN
630 LB=LEN(B$)-1
640 B$=''(+LEFT$(B$,J-1)+MID$(B$,J+1)
650 B$=LEFT$(B$,LB)
660 IF RIGHT$(B$,2)='')'' THEN B$=LEFT$(B$,LEN(B$)-2) +'')''
670 RETURN
680 REM **********
690 REM ** ATOM **
700 A$=MID$(A$,7,LEN(A$)-1)
710 J=0:B$=''NIL''
720 J=J+1
730 IF MID$(A$,J,1)='' ''OR MID$(A$,J,1)=''('' THEN RETURN
740 IF J<LEN(A$) THEN 720
750 B$=''T''
```

```
760 RETURN
770 REM **********
780 REM ** EQ **
790 A$=MID$(A$,5)
800 A$=LEFT$(A$,LEN(A$)-1)
810 J=0:B$=''NIL''
820 J=J+1
830 IF MID$(A$,J,1)='')'' THEN RETURN
840 IF MID$(A$,J,1)='' '' THEN 870
850 IF J<LEN(A$) THEN 820
860 RETURN
870 C$=LEFT$(A$,J-1)
880 A$=MID$(A$,J+1)
890 IF C$=A$ THEN B$=''T''
900 RETURN
910 REM **********
920 REM ** NULL **
930 B$=''NIL''
940 IF A$=''NULL'' ()'' THEN PRINT ''ILLEGAL - NULL NEEDS
    ARGUMENT'':B$='' ''
950 IF A$=''NULL (())'' THEN PRINT B$=''T''
960 RETURN
```

15

SSLISP

Usando o EASLE o leitor aprendeu como usar as seis funções básicas da LISP, a saber, a CAR, a CDR, a ATOM, a CONS, a EQ e a NULL. No nosso principal programa tipo LISP, o SSLISP (de «Single Statement LISP»), teremos acesso a estas seis funções e muitas outras (31 no total) geralmente existentes nos sistemas LISP.

A SSLISP permite-nos até definir as nossas próprias funções, pelo que se pensarmos que PRIMEIRO é um nome mais lógico do que CAR poderemos definir a função PRIMEIRO que se comportará exactamente do mesmo modo que a CAR. A SSLISP permite-nos definir até vinte funções novas.

Como as seis funções originais são exactamente as mesmas em SSLISP e em EASLE, não apresentarei aqui exemplos do seu uso. Em vez disso, concentrar-nos-emos nas novas funções.

MEMBER

MEMBER procura a primeira lista no interior da lista estudada, e produz a lista cuja primeira lista é o CAR. Se não encontra uma lista nestas condições, produz NIL. MEMBER utiliza EQUAL (a que chegaremos daqui a pouco) na sua definição, enquanto a função que se segue — MEMQ — utiliza EQ. Em alguns sistemas MEMBER testa a presença de um átomo na lista, produzindo T ou NIL.

```
: MEMBER ((FIM) (NO (FIM) (FAZEMOS A ASSINATURA)))
   VALUE IS...
   ((FIM) (FAZEMOS A ASSINATURA))

: MEMBER (ESTE ((ESTE TESTE) PROVA-O))
   VALUE IS...
   NIL
```

MEMQ

Como disse anteriormente, MEMQ utiliza EQ na sua definição tal como MEMBER usa EQUAL. O exemplo de execução seguinte mostra o efeito desta alteração na prática:

```
: MEMQ (AGORA (TALVEZ AGORA SEJA MELHOR))
  VALUE IS..
  (AGORA SEJA MELHOR)

: MEMQ (AGORA (TALVEZ (AGORA) SEJA MELHOR))
  VALUE IS...
  NIL
```

APPEND

Esta função é usada para criar uma lista única por junção de duas outras. O exemplo de execução que se segue mostra o seu comportamento:

```
: APPEND ((VAMOS TENTAR) (JUNTAR DUAS LISTAS))
  VALUE IS...
  (VAMOS TENTAR JUNTAR DUAS LISTAS)

: APPEND ((JUNTA DUAS COISAS) (NUMA SÓ))
  VALUE IS...
  (JUNTA DUAS COISAS NUMA SÓ)
```

REVERSE

Esta função é ideal para inverter os elementos de uma lista, desde que não se encontrem dentro de parêntesis próprios. O segundo exemplo esclarecerá melhor esta limitação:

```
: REVERSE ((PARA BAIXO E PARA TRÁS))
  VALUE IS...
  (TRÁS PARA E BAIXO PARA)

: REVERSE ((NOVE POUPA (UM REMENDO (A TEMPO))))
  VALUE IS...
  ((UM REMENDO (A TEMPO)) POUPA NOVE)
```

EQUAL

Já mencionámos EQUAL ao falarmos de MEMBER e MEMQ. EQUAL verifica se as duas partes da lista examinada são idênticas. Os dois argumentos desta função podem ser, como se pode ver, números, listas ou átomos:

```
: EQUAL ((ESTE) (ESTE))
  VALUE IS...
  T

: EQUAL (9 9)
  VALUE IS...
  T

: EQUAL ((ESTE (É)) (ESTE (É)))
  VALUE IS...
  T

: EQUAL ((ESTE) (AQUELE))
  VALUE IS...
  NIL

: EQUAL (9 -9)
  VALUE IS...
  NIL
```

LIST

Enquanto EQUAL, e muitas outras funções LISP, pode aceitar dois e apenas dois argumentos, LIST aceitará qualquer número destes. Produz os valores dos argumentos da função (o que significa que apenas produz os valores da lista):

```
: LIST (PODEMOS DESCOBRIR TUDO)
  VALUE IS...
  (PODEMOS DESCOBRIR TUDO)

: LIST (PODEMOS (TALVEZ DESCOBRIR) TUDO)
  VALUE IS...
  (PODEMOS (TALVEZ DESCOBRIR) TUDO)
```

```
: LIST (PODEMOS (TALVEZ (DESCOBRIR)) TUDO)
   VALUE IS..
   (PODEMOS (TALVEZ (DESCOBRIR)) TUDO)

: LIST (PODEMOS (TALVEZ (DESCOBRIR (TUDO))))
   VALUE IS...
   (PODEMOS (TALVEZ (DESCOBRIR (TUDO))))
```

Operações matemáticas

Vimos já as estranhas contorções necessárias em PROLOG para executar funções matemáticas. A LISP coloca-nos em situações semelhantes como veremos em seguida.

O primeiro par de funções que estudaremos são ADD1 e SUB1, que somam e subtraem 1 ao argumento único que se segue a qualquer delas:

```
: AAD1 (9)
   VALUE IS...
   10

: ADD1 (-100)
   VALUE IS...
   -99

: ADD1 (0)
   VALUE IS...
   1

: SUB1 (0)
   VALUE IS...
   -1

: SUB1 (-99)
   VALUE IS...
   -100
```

ZEROP verifica se o argumento da função é zero ou não, respondendo T ou N. ONEP faz o mesmo relativamente ao número 1:

```
: ZEROP (1)
   VALUE IS...
   NIL
```

```
: ZEROP (9)
  VALUE IS
  NIL

: ZEROP (0)
  VALUE IS..
  T

: ONEP (1)
  VALUE IS...
  T

: ONEP (0)
  VALUE IS...
  NIL
```

NUMBERP produzirá T se o argumento for um número, ou NIL se não o for, enquanto que a função MINUSP verifica se o argumento é ou não um número negativo:

```
: NUMBERP (NOVE)
  VALUE IS...
  NIL

: NUMBERP (9)
  VALUE IS...
  T

: MINUSP (9)
  VALUE IS...
  NIL

: MINUSP (-9)
  VALUE IS...
  T

:
```

As duas funções que se seguem, GREATERP e LESSP, exigem dois argumentos. Comparam o valor do primeiro argumento com o segundo, produzindo T ou NIL:

```
: GREATERP (999 12)
  VALUE IS...
  T
```

```
: GREATERP (12 888)
  VALUE IS...
  NIL

: LESSP (12 888)
  VALUE IS...
  T

: LESSP (-10 -1)
  VALUE IS...
  T

: LESSP (999 12)
  VALUE IS...
  NIL
```

Em seguida teremos MAX e MIN, que podem aceitar qualquer número de argumentos. Produzem o valor máximo e mínimo existente nas respectivas listas:

```
: MAX (9 -8 123 0 33 19)
  VALUE IS...
  123

: MIN (9 -8 123 0 33 19)
  VALUE IS...
  -8
```

MINUS altera o sinal de um argumento único (multiplicando-o de facto por menos um):

```
: MINUS (8)
  VALUE IS...
  -8

: MINUS (-8)
  VALUE IS...
  8
```

Trabalhando numericamente

Enquanto as funções matemáticas acima são essencialmente «passivas», produzindo uma indicação sobre os argumentos ou alterando um número de apenas uma unidade, o conjunto seguinte é usado para realizar de facto opeações matemáticas.

DIFFERENCE produz a diferença de valor entre os dois argumentos, subtraindo o segundo do primeiro:

```
: DIFFERENCE (88 12)
  VALUE IS...
  76

: DIFFERENCE (100 -100)
  VALUE IS...
  200
```

Pode-se elevar o primeiro argumento à potência do segundo usando EXPT:

```
: EXPT (3 3)
  VALUE IS...
  27

: EXPT (3 -3)
  VALUE IS...
  3.703704E-02
```

RECIP produz o valor da divisão de um pelo argumento:

```
: RECIP (3)
  VALUE IS...
  .3333334

: RECIP (10)
  VALUE IS...
  .1

: RECIP (.1)
  VALUE IS...
  10
```

QUOTIENT divide o primeiro argumento pelo segundo:

```
: QUOTIENT (12 3)
  VALUE IS...
  4

: QUOTIENT (3 12)
  VALUE IS...
  .25
```

REMAINDER produz o resto da divisão aritmética (em muitos BASIC's o operador MOD produz o valor inteiro que constitui o resto de uma divisão inteira; REMAINDER é o equivalente LISP desta função):

```
: REMAINDER (01 3)
  VALUE IS...
  1

: REMAINDER (100 9)
  VALUE IS...
  1

: REMAINDER (99 11)
  VALUE IS...
  0

: REMAINDER (12)
  VALUE IS...
  ERROR - ONLY ONE ARGUMENT

: REMAINDER (12 3)
  VALUE IS...
  0
```

Enquanto muitas das funções acima só podem tratar dois argumentos, as duas últimas que citaremos, PLUS e TIMES, aceitarão qualquer número de argumentos:

```
: PLUS (8 12 -3)
  VALUE IS...
  17

: PLUS (-8 -12 3)
  VALUE IS...
  -17

: TIMES (12 3)
  VALUE IS...
  36

: TIMES (12 3 3)
  VALUE IS...
  108
```

```
: TIMES (24   .01)
   VALUE IS...
   24

: TIMES (24  .1)
   VALUE IS...
   2.4
```

16

DEFINIÇÃO DE FUNÇÕES EM LISP

Já mencionei, ao discutir as seis funções básicas disponíveis nos sistemas LISP (CAR, CDR, etc.), o facto de estas poderem ser usadas para definir quaisquer funções que se queira criar num sistema computador. A LISP utiliza a função DEFINE para definir novas funções. A nossa função DEFINE segue a sintaxe normal da LISP para as funções definidas pelo utilizador, pelo que a prática adquirida usando a SSLISP pode servir bastante para um verdadeiro sistema LISP. No entanto, como a SSLISP só pode aceitar uma declaração de cada vez, estamos aqui limitados à alteração do nome de uma função suportada pelo sistema. Podemos assim alterar os nomes de um máximo de vinte funções existentes na SSLISP.

A sintaxe para o uso da DEFINE é a seguinte:

```
DEFINE (NOVO-NOME (LAMBDA (VAR-FALSA) (FUNÇÃO VAR-FALSA)))
```

Em comum com praticamente tudo o que existe no mundo da LISP, a definição de uma função é uma lista. O CAR da lista é o nome da função (NOVO-NOME é o exemplo dado). Este nome pode ser qualquer átomo, mas não deve ser um nome que esteja a ser usado pelo sistema como função. O novo nome é seguido pela palavra LAMBDA (vindo este nome do formalismo lógico chamado «lambda-calculus» desenvolvido e explicado por Alonso Church na sua *Introduction to Mathematical Logic,* vol. 1; Princeton, Nova Jersey; Princeton University Press, 1956). Apesar da sua impressionante ascendência, podemos efectivamente ignorar a palavra, sabendo no entanto que deve existir para que a função DEFINE funcione.

Depois de LAMBDA encontramos uma variável falsa (do tipo que será utilizado pela palavra definida). Em seguida está a função de sistema que desejamos imitar (nos verdadeiros sistemas LISP podem-se utilizar outras definições neste ponto, conduzindo a definições complexas e muito potentes. A FORTH garante a mesma potência). O último elemento da lista é de novo a variável falsa.

Isto pode parecer tudo um pouco estranho (julgo aliás que é um dos elementos mais difíceis de compreender na LISP). No entanto, de-

pois de vermos como se definem funções na prática compreendemos que é bastante simples.

A função CAR, como o leitor já sabe, produz o primeiro elemento de uma lista. O nosso primeiro uso de DEFINE será criar uma função chamada FIRST («primeiro») que produza o CAR de uma lista. Escrevemos o seguinte:

```
: DEFINE (FIRST (LAMBDA (FALSA) (CAR FALSA)))
  VALUE IS...
  FIRST
```

Verificamos depois a função usando FIRST como se fosse uma das instruções standard do sistema:

```
: FIRST ((ISTO É UM TESTE))
  VALUE IS...
  ISTO

: FIRST (((ISTO É) UM TEXTE))
  VALUE IS...
  (ISTO É)
```

Como vemos, funciona. Se quisermos usar um X em vez da palavra TIMES que executa multiplicações, podemos DEFINE X do seguinte modo:

```
: DEFINE (X (LAMBDA (NUM) (TIMES NUM)))
  VALUE IS...
  X

: X (8 9)
  VALUE IS...
  72
```

Podemos usar BIGGEST para funcionar como MAX:

```
: DEFINE (BIGGEST (LAMBDA (JJ) (MAX JJ)))
  VALUE IS...
  BIGGEST

: BIGGEST (1 44 -9 182 12)
  VALUE IS...
  182
```

Finalmente, acrescentaremos BOMB ao nosso vocabulário, funcionando como ATOM:

```
: DEFINE (BOMB (LAMBDA (LIST) (ATOM LIST)))
  VALUE IS...
  BOMB

: BOMB (HAHA)
  VALUE IS...
  T
```

17

LISTAGEM DA SSLISP

Vejamos a listagem completa da SSLISP:

```
10 REM S.S. LISP
20 GOTO 3450:REM INICIALIZAR
30 REM **********
40 A$=MID$(A$,E):A$=LEFT$(A$,LEN(A$)-1)
50 RETURN
60 REM **********
70 PRINT
80 NN=0
90 INPUT '': '',A$
100 IF A$='' '' THEN END:REM CARREGAR EM ''RETURN'' PARA TERMINAR
110 REM **********
120 FOR J=1 TO 12
130 X(J)=0:Y(J)=0:Z(J)=0
140 NEXT J
150 REM **********
160 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
170 FOR J=1 TO LEN(A$)
180 B$=MID$(A$,J,1)
190 IF B$=''(''THEN S=S+1:Z(S)=J:IF T=0 THEN CFIRST=J
200 IF B$='')'' THEN T=T+1:Y(T)=J:IF CSECND<>0 AND EDGE=0 THEN
    EDGE=J-1
210 IF T=1 AND B$='')'' THEN CSECND=J
220 IF B$='' '' THEN R=R+1:X(R)=J
230 NEXT J
240 IF S=T THEN 300:REM EQUILÍBRIO ( )
250 IF S<T THEN PRINT ''-> MISSING (''
260 IF S>T THEN PRINT ''-> MISSING )''
270 INPUT''  +    '',B$
280 A$=A$+B$
290 GOTO 120
300 IF NWDS=0 OR NN=1 THEN 370
310 M$=LEFT$(A$,X(1)-1)
```

```
320 FOR J=1 TO NWDS
330 IF M$=N$(J) THEN A$=O$(J)+MID$(A$,LEN(N$(J))+1)
340 NEXT J
350 NN=1
360 GOTO 120
370 FLAG=0:B$=''NIL''
380 IF LEFT$(A$,5)=''CAR ('' THEN FLAG=1
390 IF LEFT$(A$,5)=''CDR ('' THEN FLAG=2
400 IF LEFT$(A$,6)=''CONS ('' THEN FLAG=3
410 IF LEFT$(A$,6)=''ATOM ('' THEN FLAG=4
420 IF LEFT$(A$,4)=''EQ ('' THEN FLAG=5
430 IF LEFT$(A$,6)=''NULL ('' THEN FLAG=6
440 IF LEFT$(A$,8)=''MEMBER ('' THEN FLAG=7
450 IF LEFT$(A$,5)=''MEMQ ''THEN FLAG=8
460 IF LEFT$(A$,8)=''APPEND ('' THEN FLAG=9
470 IF LEFT$(A$,9)=''REVERSE ('' THEN FLAG=10
480 IF LEFT$(A$,7)=''EQUAL ('' THEN FLAG=11
490 IF LEFT$(A$,6)=''LIST (''THEN FLAG=12
500 IF LEFT$(A$,8)=''DEFINE ('' THEN FLAG=13
510 IF LEFT$(A$,6)=''ADD1 ('' THEN FLAG=14
520 IF LEFT$(A$,6)=''SUB1 ('' THEN FLAG=15
530 IF LEFT$(A$,7)=''ZEROP ('' THEN FLAG=16
540 IF LEFT$(A$,12)=''DIFFERENCE ('' THEN FLAG=17
550 IF LEFT$(A$,6)=''EXPT ('' THEN FLAG=18
560 IF LEFT$(A$,5)=''MAX ('' THEN FLAG=19
570 IF LEFT$(A$,5)=''MIN ('' THEN FLAG=20
580 IF LEFT$(A$,6)=''PLUS ('' THEN FLAG=21
590 IF LEFT$(A$,7)=''MINUS ('' THEN FLAG=22
600 IF LEFT$(A$,10)=''QUOTIENT ('' THEN FLAG=23
610 IF LEFT$(A$,7)=''RECIP ('' THEN FLAG=24
620 IF LEFT$(A$,11)=''REMAINDER ('' THEN FLAG=25
630 IF LEFT$(A$,7)=''TIMES ('' THEN FLAG=26
640 IF LEFT$(A$,10)=''GREATERP ('' THEN FLAG=27
650 IF LEFT$(A$,7)=''LESSP ('' THEN FLAG=28
660 IF LEFT$(A$,8)=''MINUSP ('' THEN FLAG=29
670 IF LEFT$(A$,9)=''NUMBERP ('' THEN FLAG=30
680 IF LEFT$(A$,6)=''ONEP ('' THEN FLAG=31
690 IF FLAG>13 THEN 720
700 ON FLAG GOSUB 840,890,970,1110,1200,1330,1380,1510,1700,
1760,2000,2130,3300
710 GOTO 760
720 IF FLAG>24 THEN 750
730 ON FLAG-13 GOSUB 2180,2230,2280,2350,2560,2600,2790,
    2820,2870,2920,2960
740 GOTO 760
```

```
750 ON FLAG-24 GOSUB 3030,3070,3120,3160,3200,3250,2280
760 IF FLAG<>0 THEN GOSUB 780
770 GOTO 70
780 REM ** RESPOSTA **
790 PRINT '' VALUE IS...''
800 IF B$<>''()'' THEN PRINT ''    '';B$
810 IF B$=''()'' THEN PRINT ''  NIL''
820 RETURN
830 REM **********
840 REM ** CAR **
850 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z(2))
860 IF S>2 THEN B$=MID$(A$,CFIRST, CSECND-CFIRST+1)
870 RETURN
880 REM **********
890 REM ** CDR **
900 GOSUB 840
910 LB=LEN(B$)+7
920 B$=''(''+MID$(A$,LB,EDGE-1)
930 IF RIGHT$(B$,2)=''))'' THEN B$=LEFT$(B$,LEN(B$)-1)
940 IF MID$(B$,2,1)='' '' THEN B$=''(''+MID$(B$,3)
950 RETURN
960 REM **********
970 REM ** CONS **
980 B$=MID$(A$,7,LEN(A$)-1)
990 J=0
1000 IF LEFT$ (B$,1)=''('' THEN J=1
1010 J=J+1
1020 IF MID$(B$,J,1)=''('' THEN 1050
1030 IF J<LEN(B$) THEN 1010
1040 B$='' * CONS ERROR *'': RETURN
1050 LB=LEN(B$)-1
1060 B$=''(''+LEFT$(B$,J-1)+MID$(B$,J+1)
1070 B$=LEFT$(B$,LB)
1080 IF RIGHT$(B$,2)='' )'' THEN B$=LEFT$(B$,LEN(B$)-2)+'')''
1090 RETURN
1100 REM **********
1110 REM ** ATOM **
1120 A$=MID$(A$,7,LEN(A$)-1)
1130 J=0
1140 J=J+1
1150 IF MID$(A$,J,1)='' '' OR MID$(A$,J,1)=''('' THEN RETURN
1160 IF J<LEN(A $) THEN 1140
1170 B$=''T''
1180 RETURN
1190 REM **********
```

```
1200 REM ** EQ **
1210 E=5:GOSUB 40
1220 J=0
1230 J=J+1
1240 IF MID$(A$,J,1)='')'' THEN RETURN
1250 IF MID$(A$,J,1)='' '' THEN 1280
1260 IF J<LEN(A$) THEN 1230
1270 RETURN
1280 C$=LEFT$(A$,J-1)
1290 A$=MID$(A$,J+1)
1300 IF C$=A$ THEN B$=''T''
1310 RETURN
1320 REM **********
1330 REM ** NULL **
1340 IF A$=''NULL ()'' THEN B$=''*ILLEGAL - NULL NEEDS
     ARGUMENT *''
1350 IF A$=''NULL (())''THEN B$=''T''
1360 RETURN
1370 REM **********
1380 REM ** MEMBER **
1390 C$=MID$(A$,9)
1400 J=1
1410 J=J+1
1420 IF MID$(C$,J,1)='')'' OR MID$(C$,J,1)=''('' THEN
     D$=LEFT$ (C$,J):GOTO 1450
1430 IF J<LEN(C$)THEN 1410
1440 RETURN
1450 J=LEN(D$)
1460 J=J+1
1470 IF MID$(C$,J,LEN(D$))=D$ THEN C$=LEFT$ (C$,LEN(C$)-1) :
     GOTO 1630
1480 IF J <LEN( C$) THEN 1460
1490 RETURN
1500 REM **********
1510 REM ** MEMQ **
1520 C$=MID$(A$,7)
1530 J=0
1540 J=J+1
1550 IF MID$(C$,J,1)='' '' THEN 1580
1560 IF J<LEN(A$) THEN 1540
1570 RETURN
1580 D$=LEFT$(C$,J)
1590 C$=MID$(C$,J+2)
1600 C$=LEFT$(C$,LEN(C$)-2)+'' ''
1610 J=0
```

```
1620 J=J+1
1630 IF MID$(C$,J,LEN(D$))=D$ THEN B$=''(''+MID$(C$,J):
     GOTO 1660
1640 IF J<LEN(C$) THEN 1620
1650 RETURN
1660 B$=LEFT$(B$,LEN(B$)-1)+'')''
1670 IF RIGHT$(B$,3)=''))))'' THEN B$=LEFT$(B$,LEN(B$)-1):
     GOTO 1670
1680 RETURN
1690 REM **********
1700 REM ** APPEND **
1710 B$=MID$(A$,9)
1720 B$=LEFT$(B$,Y(1)-9)+'' ''+MID$(B$,Z(3)-7)
1730 B$=LEFT$(B$,LEN(B$)-1)
1740 RETURN
1750 REM **********
1760 REM ** REVERSE **
1770 B$='' ''
1780 A$=MID$(A$,11):A$=LEFT$(A$,LEN(A$)-2)
1790 CT=0
1800 J =0
1810 J=J+1:IF J>LEN(A$) THEN 1920
1820 IF MID$(A$,J,1)='' ''THEN 1850
1830 IF MID$(A$,J,1)=''(''THEN 1860
1840 GOTO 1810
1850 CT=CT+1:G$(CT)=LEFT$(A$,J-1):A$=MID$ (A$,J+1):GOTO 1800
1860 J=J+1:IF MID$(A$,J,2)=''))'' THEN 1980
1870 IF MID$(A$,J,1)='')'' THEN 1910
1880 IF J=LEN(A$) THEN 1900
1890 GOTO 1860
1900 CT=CT+1:G$(CT)=A$+'')'':GOTO 1930
1910 CT=CT+1:G$(CT)=LEFT$(A$,J):A$=MID$(A$,J+1):GOTO 1800
1920 CT=CT+1:G$(CT)=A$
1930 FOR M=CT TO 1 STEP -1
1940 B$=B$+G$(M):IF M>1 THEN B$=B$+'' ''
1950 NEXT M
1960 B$=''(''+B$+'')''
1970 RETURN
1980 CT=CT+1:G$(CT)=LEFT$(A$,J+1):A$=MID$ (A$,J+2):GOTO 1800
1990 REM **********
2000 REM ** EQUAL **
2010 E=8:GOSUB 40
2020 M=ASC(A$):IF M>47 AND M<58 THEN 2370
2030 J=0
2040 J=J+1
```

```
2050 IF MID$(A$,J,2)='') ''THEN J=J+1:GOTO 1280
2060 IF MID$(A$,J,3)=''))) '' THEN 2100
2070 IF MID$(A$,J,2)='')) '' THEN 2110
2080 IF J<LEN(A$) THEN 2040
2090 RETURN
2100 C$=LEFT$(A$,J+2):A$=MID$(A$,J+4):GOTO 1300
2110 C$=LEFT$(A$,J+1):A$=MID$(A$,J+3):GOTO 1300
2120 REM **********
2130 REM ** LIST **
2140 E=7:GOSUB 40
2150 B$=''(''+A$+'')''
2160 RETURN
2170 REM **********
2180 REM ** ADD1 **
2190 E=7:GOSUB 40
2200 B$=STR$(VAL(A$)+1)
2210 RETURN
2220 REM **********
2230 REM ** SUB1 **
2240 E=7:GOSUB 40
2250 B$=STR$(VAL(A$)-1)
2260 RETURN
2270 REM **********
2280 REM ** ZEROP **
2290 IF FLAG=16 THEN E=8
2300 IF FLAG=31 THEN E=7
2310 GOSUB 40
2320 IF A$=''0'' AND FLAG=16 OR A$=''1'' AND FLAG=31 THEN
     B$=''T''
2330 RETURN
2340 REM **********
2350 REM ** DOIS ARGUMENTOS **
2360 E=13:GOSUB 40
2370 J=0
2380 J=J+1
2390 IF MID$(A$,J,1)='' ''THEN 2420
2400 IF J<LEN(A$) THEN 2420
2410 B$=''* ERROR - ONLY ONE ARGUMENT *'':RETURN
2420 P=VAL(LEFT$(A$,J-1))
2430 Q=VAL(MID$(A$,J+1))
2440 IF FLAG=17 THEN B$=STR$(P-Q):RETURN
2450 IF FLAG=23 OR FLAG=25 THEN B=P/Q
2460 IF FLAG=25 THEN B=INT(.5+Q*(B-INT(B))* 1000)/1000
2470 IF FLAG=18 THEN B=P^Q
2480 IF FLAG=11 AND P=Q THEN B$=''T''
```

```
2490 IF FLAG=27 AND P>Q THEN B$=''T''
2500 IF FLAG=28 AND P<Q THEN B$=''T''
2510 IF FLAG=32 THEN B=P-Q
2520 IF FLAG=11 OR FLAG>26 THEN RETURN
2530 B$=STR$(B)
2540 RETURN
2550 REM **********
2560 REM ** EXPT **
2570 E=7:GOSUB 40
2580 GOTO 2370
2590 REM **********
2600 REM ** MAX (MIN PLUS TIMES) **
2610 F$=LEFT$(A$,3):A$=MID$(A$,6)
2620 CT=0:FLAG=0
2630 IF F$=''TIMES'' THEN CT=1
2640 J=0
2650 J=J+1
2660 IF MID$(A$,J,1)='' '' THEN 2690
2670 IF J< LEN(A$) THEN 2650
2680 IF J=LEN(A$) THEN FLAG=1
2690 P=VAL(LEFT$(A$,J-1)):IF FLAG=0 THEN A$=MID$(A$,J+1)
2700 IF F$<>''PLUS'' AND CT=0 THEN CT=P
2710 IF F$=''MAX'' AND P>CT THEN CT=P
2720 IF F$=''MIN'' AND P<CT THEN CT=P
2730 IF F$=''PLUS'' THEN CT=CT+P
2740 IF F$=''TIMES'' THEN CT=CT*P
2750 IF FLAG=0 THEN 2640
2760 B$=STR$(CT)
2770 RETURN
2780 REM **********
2790 REM ** MIN **
2800 GOTO 2160
2810 REM **********
2820 REM ** PLUS **
2830 F$=''PLUS''
2840 A$=MID$(A$,7)
2850 GOTO 2620
2860 REM **********
2870 REM ** MINUS **
2880 E=8:GOSUB 40
2890 B$=STR$(-VAL(A$))
2900 RETURN
2910 REM **********
2920 REM ** QUOCIENTE **
2930 E=11:GOSUB 40
```

```
2940 GOTO 2370
2950 REM **********
2960 REM ** RECIP **
2970 E=8:GOSUB 40
2980 IF A$=''0'' THEN B$='' DIVISION BY ZERO ILLEGAL'':RETURN
2990 B=1/(VAL(A$))
3000 B$=STR$ (B)
3010 RETURN
3020 REM **********
3030 REM ** REMAINDER **
3040 E=12:GOSUB 40
3050 GOTO 2370
3060 REM **********
3070 REM ** TIMES **
3080 F$=''TIMES''
3090 A$=MID$(A$,8)
3100 GOTO 2620
3110 REM **********
3120 REM ** GREATERP **
3130 E=11:GOSUB 40
3140 GOTO 2370
3150 REM **********
3160 REM ** LESSP **
3170 E=8:GOSUB 40
3180 GOTO 2370
3190 REM **********
3200 REM ** MINUSP **
3210 E=9:GOSUB 40
3220 IF VAL(A$)<0 THEN B$=''T''
3230 RETURN
3240 REM **********
3250 REM ** NUMBERP **
3260 A$=MID$(A$,10)
3270 IF ASC(A$)>44 AND ASC(A$)<58 THEN B $=''T''
3280 RETURN
3290 REM **********
3300 REM ** DEFINE **
3310 A$=MID$(A$,9)
3320 F$=LEFT$(A$,X(2)-9)
3330 G$=MID$(A$,X(4)-6)
3340 J=0
3350 J=J+1
3360 IF MID$(G$,J,1)3'' '' THEN 3390
3370 IF J<LEN(G$) THEN 3350
3380 B$='' DEFINE ERROR'':RETURN
```

```
3390 G$=LEFT$(G$,J-1)
3400 NWDS=NWDS+1
3410 O$(NWDS)=G$:N$(NWDS)=F$
3420 B$=F$
3430 RETURN
3440 REM **********
3450 REM INICIALIZAR
3460 CLS
3470 DIM G$(20),O$(20),X(12),Y(12),Z(12), N$(20)
3480 NWDS=0: REM CONTAGEM DAS PALAVRAS NOVAS DEFINIDAS PELO
     UTILIZADOR
3490 GOTO 70
```

LISTAGENS ESPECÍFICAS PARA VÁRIOS MICROCOMPUTADORES

MECÂNICO

```
10  REM MECÂNICO
15  POKE 53280,9:POKE 53281,0
20  PRINT ''
(CLR)''
30  PRINT''TEM PORTANTO PROBLEMAS NO SEU AUTOMÓVEL.''
40  PRINT''VEJAMOS SE OS PODEMOS RESOLVER...''
50  PRINT
60  PRINT''INDIQUE A OPÇÃO QUE CORRESPONDE AO PROBLEMA:''
70  PRINT
80  PRINT''
(OR) A - O MOTOR NÃO ARRANCA''
90  PRINT''
(CYN) B - O MOTOR DE ARRANQUE FUNCIONA, MAS NÃO ARRANCA''
100 PRINT''
(PUR) C - ARRANCA MAS PÁRA IMEDIATAMENTE''
110 PRINT''
(GRN) D - FUNCIONA, MAS VAI ABAIXO MUITAS VEZES''
120 PRINT''
(BLU) E - FUNCIONA, MAS É IRREGULAR EM PONTO MORTO(WHT)''
130 GET A$:IF A$<>'' '' THEN 130
140 GET A$
150 IF A$<''A'' OR A$>''E'' THEN 140
160 GOSUB 1560
170 ON (ASC(A$)-64) GOTO 200,580,1200,1230,1350
180 END
190 REM *************************
200 REM NÃO ARRANCA
210 PRINT''
(OR)COMEÇAMOS POR VERIFICAR A BATERIA.''
220 PRINT ''LIGUE AS LUZES.''
230 PRINT TAB(6);''ESTÃO FRACAS?''
240 GOSUB 1520
```

```
250 IF A$=''N'' THEN 350:REM BATERIA OK
260 PRINT ''OS CABOS DA BATERIA ESTÃO MAL FIXOS OU
    CORROÍDOS?''
270 GOSUB 1520
280 IF A$=''S'' THEN PRINT ''APERTE E LIMPE.''
290 GOSUB 1580
300 PRINT''A CORREIA DA BATERIA ESTÁ POUCO APERTADA?''
310 GOSUB 1520
320 IF A$=''S'' THEN PRINT TAB(8);''APERTE-A''
330 GOSUB 1580
340 PRINT''UM CABO LIGADO A OUTRA BATERIA OU UM EMPURRÃO
    DEVEM BASTAR PARA ARRANCAR O MOTOR'':END
350 PRINT''O CABO DA BATERIA ESTÁ DESAPERTADO OU CORROÍDO
    JUNTO A ESTA''
370 GOSUB 1520
380 IF A$=''S'' THEN PRINT ''APERTE-O E LIMPE AS LIGAÇÕES''
390 GOSUB 1580
400 PRINT''COLOQUE UMA PEÇA METÁLICA ENTRE OS TERMINAIS DO
    SOLENÓIDE.''
410 PRINT''O MOTOR DE ARRANQUE FUNCIONA?''
420 GOSUB 1520
430 IF A$=''S'' THEN PRINT''A CHAVE DE IGNIÇÃO ESTÁ
    PROVAVELMENTE DEFEITUOSA''
440 GOSUB 1580
450 IF A$=''S'' THEN PRINT TAB(4);''DEVE SER
    SUBSTITUÍDA'':END
460 PRINT''O MOTOR DE ARRANQUE PARECE QUERER ARRANCAR?''
470 GOSUB 1520
480 IF A$=''S'' THEN PRINT ''O MOTOR DE ARRANQUE PODE ESTAR
    PRESO'':GOTO 520
490 PRINT''COMO NADA ACONTECEU, O PROBLEMA''
500 PRINT''RESIDE PROVAVELMENTE NO SOLENÓIDE''
510 PRINT''UM EMPURRÃO DEVERÁ ARRANCAR O MOTOR.'':END
520 REM MOTOR DE ARRANQUE PRESO
530 PRINT''DESLIGUE A IGNIÇÃO, E COLOQUE O MOTOR EM''
540 PRINT''QUARTA VELOCIDADE. EMPURRE O CARRO UM POUCO''
550 PRINT''PARA O LIBERTAR.'':END
560 RETURN
570 REM ********************
580 REM MOTOR ELÉCTRICO RODA, MAS NÃO ARRANCA
590 PRINT''
(CYN) VERIFIQUE SE HÁ HUMIDADE NAS VELAS.''
600 PRINT''ENCONTRA ÁREAS HÚMIDAS?''
610 GOSUB 1520
620 IF A$=''S'' THEN PRINT''UTILIZE UM PULVERIZADOR CONTRA A
```

```
    HUMIDADE'':GOSUB 1580
630 PRINT''VÊ POEIRAS NA PARTE ISOLANTE DA MOLA''
640 PRINT TAB(6);''OU NA TAMPA DO DISTRIBUIDOR?''
660 GOSUB 1520
670 IF A$=''S'' THEN PRINT''LIMPE A BOBINA E O INTERIOR''
680 IF A$=''S'' THEN PRINT''E EXTERIOR DA TAMPA. '':GOSUB
    1580
690 PRINT''VERIFIQUE SE TODOS OS FIOS ESTÃO APERTADOS ANTES
    DE CONTINUAR''
700 GOSUB 1580
710 PRINT ''SE O MOTOR NÃO ARRANCA, DEVE VERIFICAR AS
    VELAS:''
720 GOSUB 1580
730 PRINT''A FAÍSCA NA EXTREMIDADE DO CABO SALTA MUITO?''
740 GOSUB 1520
750 IF A$=''S'' THEN PRINT ''AS VELAS ESTÃO DEFEITUOSAS'':
    GOSUB 1580
760 IF A$=''N'' THEN 830
770 PRINT''VÊ GORDURA NAS VELAS?''
780 GOSUB 1520
790 IF A$=''S'' THEN PRINT ''NÃO É POSSÍVEL UM ARRANJO COM AS
    VELAS NESSE ESTADO''
800 IF A$=''S'' THEN GOSUB 1580:GOTO 770
810 IF A$=''N'' PRINT ''SE SE TRATA DE UMA EMERGÊNCIA, TENTE
    DIMI-''
820 IF A$=''N'' THEN PRINT''NUIR A FOLGA PARA CERCA DE METADE
    DO NORMAL'':GOSUB 1580:GOTO 910
830 PRINT ''OBSERVA FAÍSCA?''
840 GOSUB 1520
850 IF A$=''S'' THEN 770
860 GOSUB 870:END
870 PRINT''VERIFIQUE O ROTOR, A MOLA E A TAMPA DO
    DISTRIBUIDOR''
880 PRINT''SE NÃO ENCONTRAR RACHAS É PROVÁVEL QUE''
890 PRINT''TENHA DE RECORRER A UM MECÂNICO'':RETURN
910 PRINT''TEM GASOLINA NO DEPÓSITO?''
920 GOSUB 1520
930 IF A$=''N'' THEN PRINT''ENCHA-O E TENTE DE NOVO'':GOSUB
    1580
940 PRINT''OS TUBOS DE GASOLINA E AR ESTÃO BEM FIXOS?''
950 GOSUB 1520
960 IF A$=''N'' THEN PRINT''FIXE-OS E TENTE DE NOVO'':GOSUB
    1580
970 PRINT''  REMOVA O FILTRO DE AR DO CARBURADOR''
980 GOSUB 1580
```

```
990 PRINT''PARECE-LHE SECO?''
1000 GOSUB 1520
1010 IF A$=''N'' THEN 1080
1020 PRINT''RODE MANUALMENTE O MOTOR ALGUMAS VEZES''
1030 PRINT''VEDANDO A ENTRADA DE AR'':GOSUB 1580
1040 PRINT''A MÃO ESTÁ HÚMIDA DE GASOLINA?''
1050 GOSUB 1520
1060 IF A$=''N'' THEN PRINT''RETIRE A TAMPA DA GASOLINA SE A
     ENTRADA''
1070 IF A$=''N'' THEN PRINT''DE AR ESTÁ VEDADA'':GOSUB
     900:END
1080 PRINT''TENTOU ARRANCAR REPETIDAMENTE NOS ÚLTIMOS
     MINUTOS?''
1100 GOSUB 1520
1110 IF A$:=''N'' THEN 1150
1120 PRINT''ESPERE CERCA DE UM MINUTO, E DEPOIS''
1130 PRINT''CARREGUE NO ACELERADOR A FUNDO E TENTE''
1140 PRINT''ARRANCAR O MOTOR''.END
1150 PRINT''O MOTOR PODE ESTAR AFOGADO.''
1160 GOSUB 1580
1170 PRINT''DÊ UMA PANCADA NO CARBURADOR PARA LIBER-''
1180 PRINT''TAR A VÁLVULA E RECOMECE.''
1190 REM ********************
1200 REM ARRANCA MAS PÁRA
1210 PRINT''
(PUR)SUGERE UMA RESISTÊNCIA DEFEITUOSA QUE DEVE SER
     SUBSTITUIDA'':END
1120 REM
1230 REM RODA MAS VAI ABAIXO
1240 PRINT''
(GRN)O PROBLEMA É CAUSADO POR:''
1250 PRINT TAB(7);''CABOS EM CURTO-CIRCUITO OU''
1255 PRINT TAB(7);''MAL APERTADOS;''
1260 PRINT TAB(7);''UMA FAÍSCA FRACA; OU''
1270 PRINT TAB(7);''UMA DEFICIÊNCIA NO SISTEMA DE''
1275 PRINT TAB(7);''ALIMENTAÇÃO''
1280 PRINT''VERIFIQUE PRIMEIRO OS CABOS'':GOSUB 1580
1290 PRINT''SE NÃO ESTIVEREM BONS, REPARE-OS. SE NÃO''
1295 PRINT''PODERÃO SER AS VELAS.'':GOSUB 870
1310 PRINT''PODE AINDA VERIFICAR AS VELAS'':GOSUB 1580
1330 GOTO 1360
1340 REM ********************
1350 REM RODA, PONTO MORTO IRREGULAR
1360 PRINT''
(BLU)PODE ACONTECER QUE UMA OU MAIS VELAS''
```

```
1370 PRINT''ESTEJAM DEFEITUOSAS.'':GOSUB 1580
1380 PRINT''DESLIGUE-AS UMA DE CADA VEZ. AS QUE NÃO''
1390 PRINT''PROVOCAM A FALHA DO MOTOR ESTÃO DEFEI-''
1400 PRINT''TUOSAS. VERIFIQUE-AS E VOLTE AO
     SISTEMA.'':GOSUB 1580
1420 PRINT''ESTE TESTE REVELOU VELAS DEFEITUOSAS?''
1430 GOSUB 1520
1440 IF A$=''S'' THEN PRINT''SUBSTITUA TODAS AS VELAS, OU
     PELO MENOS''
1450 IF A$=''S'' THEN PRINT''AS DEFEITUOSAS'':GOSUB 1580
1460 PRINT''A CAUSA MAIS COMUM DE UM MAU PONTO MORTO''
1470 PRINT''NO CASO DE AS VELAS ESTAREM BOAS, É UMA''
1480 PRINT''MÁ MISTURA DO COMBUSTÍVEL (MUITO RICA)''
1490 PRINT''DEVE PORTANTO AJUSTAR A MISTURA'':END
1500 PRINT''AJUSTE O JIGLER NO CARBURADOR'':END
1510 REM ********************
1520 PRINT TAB(16);(S - SIM, N - NÃO)?''
1530 GET A$:IF A$<>'' '' THEN 1530
1540 GET A$
1550 IF A$<>''S'' AND A$<>''N'' THEN 1540
1560 PRINT TAB(22);''>  OK   '';A$;''<''
1570 REM ********************
1580 FOR T=1 TO 1000:NEXT T
1590 PRINT
1600 RETURN
```

MEDICI

```
10  REM MEDICI
20  POKE 53280,0:POKE 53281,0:PRINT''
(CLR)(WHT)''
30  GOSUB 1250
40  PRINT''
(PUR)A - SOU BASTANTE GORDO''
50  PRINT''B - SOU RELATIVAMENTE GORDO''
60  PRINT''C - TENHO ALGUM PESO A MAIS''
70  PRINT''D - TENHO O PESO CERTO''
80  PRINT''E - SOU MAIS MAGRO DO QUE DEVERIA (WHT)''
90  GOSUB 1170
100 WEIGHT=5*(ASC(A$)-68):IF A$=''E'' THEN WEIGHT=0
110 PRINT WEIGHT
120 GOSUB 1250
130 PRINT''FAÇO EXERCÍCIO, ELEVANDO O RITMO DO CO-''
140 PRINT''RAÇÃO PARA 120 OU MAIS, PELO MENOS O''
```

```
150 PRINT''SEGUINTE NÚMERO DE HORAS POR SEMANA:''
170 PRINT
180 PRINT ''
(CYN)A - MENOS DE UM QUARTO DE HORA''
190 PRINT''B - MAIS DE UM QUARTO, ATÉ TRÊS QUARTOS DE HORA''
200 PRINT''C - DE TRÊS QUARTOS DE HORA ATÉ HORA E MEIA''
210 PRINT''D - ENTRE UMA HORA E MEIA E DUAS E MEIA''
220 PRINT''E - MAIS DE DUAS HORAS E MEIA (WHT)''
230 GOSUB 1170
240 EXERCISE=5*(ASC(A$)-63)-5:IF A$=''A'' THEN EXERCISE=0
250 PRINT EXERCISE
260 GOSUB 1250
270 PRINT''AO CONDUZIR:'':PRINT
280 PRINT''
(BLU)A - NUNCA USO O CINTO DE SEGURANÇA''
290 PRINT''B - USO O CINTO CERCA DE UM QUARTO DAS VEZES''
300 PRINT''C - USO O CINTO EM UMA DE CADA DUAS VIAGENS''
310 PRINT''D - USO O CINTO QUASE SEMPRE''
320 PRINT''E - USO SEMPRE O CINTO DE SEGURANÇA(WHT)''
330 GOSUB 1170
340 SEATBELT=2*ASC(A$)-65)
350 PRINT SEATBELT
360 GOSUB 1250
370 PRINT ''PREOCUPO-ME COM A NUTRIÇÃO E TENTO COMER DE MODO
    SALUTAR:''
380 PRINT
390 PRINT''
(GRN)A - SEMPRE
400 PRINT''B - QUASE SEMPRE''
410 PRINT''C - UMA PROPORÇÃO RAZOÁVEL DAS VEZES''
420 PRINT''D - DE VEZ EM QUANDO''
430 PRINT''E - QUASE NUNCA
(WHT)''
440 GOSUB 1170
450 DIET=-ASC(A$)+69
460 PRINT DIET
470 GOSUB 1250
480 PRINT''FUMAR(UM CHARUTO CONTA COMO UM CIGARRO)''
490 PRINT
500 PRINT''
(OR)A - NUNCA''
510 PRINT''B - MENOS DE QUINZE CIGARROS POR DIA''
520 PRINT''C - 15 A 25 CIGARROS POR DIA''
530 PRINT''D - 26 A 42 CIGARROS POR DIA''
540 PRINT''E - MAIS DE 42 CIGARROS POR DIA
```

```
(WHT)''
550 GOSUB 1170
560 SMOKING=-7*(ASC(A$)-65)
570 PRINT SMOKING
580 GOSUB 1250
590 PRINT ''ALCOOL - QUANTAS BEBIDAS EM MÉDIA POR DIA?''
600 PRINT
610 PRINT''
(LT RED)A - NENHUMA''
620 PRINT''B - MENOS DE 3''
630 PRINT''C - 3 A 6''
640 PRINT''D - 7 A 9''
650 PRINT''E - MAIS DE 9
(WHT)''
660 GOSUB 1170
670 DRINK=-30
680 IF A$=''A'' THEN DRINK=0
690 IF A$=''B'' THEN DRINK=1
700 IF A$=''C'' THEN DRINK=DRINK/5
710 IF A$=''D'' THEN DRINK=DRINK/2
720 PRINT DRINK
730 GOSUB 1250
740 PRINT''EM GERAL, QUAL O STRESS DA SUA VIDA''
750 PRINT''NOS ÚLTIMOS 6 MESES''
760 PRINT
770 PRINT''
(LT GRN)A - UM STRESS EXTREMO''
780 PRINT''B - BASTANTE STRESS''
790 PRINT''C - UM STRESS RELATIVO''
800 PRINT''D - NEUTRO''
810 PRINT''E - SEM STRESS
(WHT)''
820 GOSUB 1170
830 SRESS=INT(2.5*(ASC(A$)-69))
840 PRINT SRESS
850 GOSUB 1300:PRINT''
(CLR)''
860 PRINT''OPINIÃO DE MEDICI:''
870 PRINT
880 PRINT TAB(8);''PESO: '';WEIGHT
890 PRINT TAB(6);''EXERCÍCIO: '';EXERCISE
900 PRINT TAB(4);''SEGURANÇA DE CONDUÇÃO: '';SEATBELT
910 PRINT TAB(5);''NUTRIÇÃO: '';DIET
920 PRINT TAB(7);''FUMO: '';SMOKING
930 PRINT TAB(7);''ÁLCOOL: '';DRINK
```

```
940 PRINT TAB(8);''STRESS: ''SRESS
950 GOSUB 1300
960 ANT=WEIGHT+EXERCISE+SEATBELT+DIET+SMOKING+DRINK+SRESS
970 GOSUB 1300:PRINT
980 PRINT''  O AMIGO VALE'';ANT:PRINT
990 PRINT''  NUMA ESCALA EM QUE A MÉDIA É ZERO,''
1000 PRINT''O MÍNIMO É INFERIOR A -80, E''
1010 PRINT''O MÁXIMO É MAIS DE 30''
1020 GOSUB 1300:PRINT
1030 IF ANT<6 AND ANT>-6 THEN A$=''MÉDIA'':
     L$=''60 A 66   65 A 71''
1040 IF ANT<-5 AND ANT>-21 THEN A$=''ABAIXO DA MÉDIA'':
     L$=''60 A 66   65 A 71''
1050 IF ANT<-20 THEN A$=''FRACO'':
     L$=''60 OU MENOS   65 OU MENOS''
1060 IF ANT<-45 THEN A$='''MUITO FRACO''
1070 IF ANT<-60 THEN A$=''MUITO, MUITO FRACO''
1080 IF ANT>5 AND ANT< 15 THEN A$=''BOM'':
     L$=''74 A 80   79 A 85''
1090 IF ANT>14 THEN A$=''EXTREMAMENTE BOM'':
     L$=''81 OU MAIS   86 OU MAIS''
1100 PRINT ''ESTADO DE SAÚDE: '';A$
1110 PRINT
1120 PRINT''EXPECTATIVA DE VIDA:''
1130 PRINT TAB(3);''HOMEM   MULHER''
1140 PRINT TAB(3);L$
1150 END
1160 REM ********************
1170 REM ACEITAR ENTRADA
1180 GET A$:IF A$<>'' ''THEN 1180
1190 GET A$
1200 IF A$<''A'' OR A$>''E'' THEN 1190
1210 PRINT:PRINT TAB(12);''OK  '';A$
1220 RETURN
1230 REM ********************
1240 REM ATRASO/ESPAÇAR
1250 FOR J=1 TO 1000:NEXT J
1260 PRINT ''
(CLR)''
1270 PRINT:PRINT:PRINT:PRINT
1280 PRINT''QUAL DAS OPÇÕES SEGUINTES ESTÁ MAIS PERTO DA
     REALIDADE:''
1290 PRINT
1300 FOR J=1 TO 400:NEXT J
1310 RETURN
```

RITA

```
10  REM RITA
20  GOSUB 1380:REM INICIALIZAR
30  GOSUB 1160:REM OPÇÕES DE SAÍDA
40  GOSUB 1250:REM OPÇÕES DE DISCRIMINAÇÃO
50  GOSUB 140:REM PERGUNTA AO UTILIZADOR
60  GOSUB 480: REM TOMAR DECISÃO E ACTUALIZAR BASE DE
    CONHECIMENTOS
65  GOSUB 2000:REM SOM
70  PRINT''
(PUR)<RETURN>PARA CONTINUAR(WHT)'';
80  IF X$<>'' '' THEN INPUT I$:GOTO 50
90  PRINT''
(PUR) APRENDIZAGEM''
100 PRINT''OU OUTRA TECLA PARA USAR RITA
(WHT)'';
110 INPUT X$:GOTO 50
120 END
130 REM ********************
140 REM PERGUNTAR A UTILIZADOR
150 PRINT''
(CLR)''
160 PRINT''
(PUR)EIS OS OBJECTOS QUE SEI DISCRIMINAR:'
170 PRINT
180 FOR J=1 TO TT
190 PRINT ''>  '';A$(J)
200 NEXT J:PRINT''
(WHT)''
210 GOSUB 1500
220 IF X$='' '' THEN PRINT''
(PUR)PENSE NUM, E CARREGUE EM <RETURN>
(WHT)''
230 IF X$<>'' '' THEN PRINT''
(PUR)ESTOU PRONTA A RESPONDER(WHT)''
240 IF X$='' '' THEN INPUT X$
250 ADD=.5
260 FOR J=1 TO DQ
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X$<>'' ''AND TT>2 THEN 390:REM VERIFICAR SE A
    PERGUNTA PODE SER OMITIDA
300 PRINT''
(PUR) ESCREVA UM NÚMERO ENTRE''
```

```
310 PRINT''1 (VERDADEIRO) E 0 (FALSO) ($ PARA TERMI-''
315 PRINT''NAR''
320 PRINT:PRINT E$(J);''
(WHT)''
330 INPUT H$:IF H$=''$'' THEN PRINT:PRINT''
(PUR) OBRIGADO (WHT)'':PRINT:END
340 C(J)=VAL(H$)
350 C(J)=ADD*C(J)
360 NEXT J
370 RETURN
380 REM ********************
390 REM VERIFICAR SE PODE OMITIR PERGUNTA
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUMP=0
430 NEXT W
440 IF JUMP=0 THEN 300
450 C(JUMP)=B(W,J)
460 GOTO 360
470 REM ********************
480 REM TOMAR DECISÃO
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXT J
520 ADD=.5
530 FOR J=1 TO TT
540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM JOGAR COM VALORES DAS TRÊS LINHAS SEGUINTES PARA
    MELHORES RESULTADOS
570 IF C(X)=B(J,X) THEN D(J)=D(J)+1
580 IF ABS(C(X)-B(J,X))<.6*ADD THEN E(J)=E(J)+.4
590 IF ABS(C(X)-B(J,X))<1.2*ADD THEN F(J)=F(J)+.1
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
690 REM * ANUNCIAR RESULTADO *
700 PRINT
710 CFLG=0
```

```
720 PRINT''
(PUR)O RESULTADO MAIS PROVÁVEL É'';A$(A1)
730 IF A2<>A1 THEN PRINT''O MAIS PROVÁVEL SEGUINTE É''
    ;A$(A2):CFLAG=1
740 IF A3 <>A2 AND A3<>A1 THEN PRINT''O MAIS PROVÁVEL A
    SEGUIR É'';A$(A3):CFLAG=2
750 PRINT
760 PRINT''O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
                (S OU N)
(WHT)'';
770 INPUT F$:F$=RIGHT$(F$,1)
780 IF F$<>''S'' AND F$<>''N'' THEN 770
790 IF F$<>''S'' AND X$<>'' ''THEN RETURN
800 IF F$=''S'' THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 980
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CFLG=0 THEN 890
840 PRINT''
(PUR)A SEGUNDA OPÇÃO ESTÁ CORRECTA
                (S OU N)(WHT)'';
850 INPUT F$:F$=RIGHT$(F$,1)
860 IF F$=''N'' THEN 890
870 IF CFLG=1 THEN A1=A2:GOTO 980
880 IF CFLG=2 THEN A1=A3:GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;''-  '';A$(J)
920 NEXT J
930 PRINT
940 PRINT''
(PUR)QUAL É A CORRECTA(WHT)''
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM ** EDUCANDO RITA **
    (ACTUALIZAR BASE DE CONHECIMENTOS)
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J)+5*B (A1,J))/6
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$='' '' THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
```

```
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K);B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
1150 REM ********************
1160 REM OPÇÕES DE SAÍDA
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
1200 PRINT''
(PUR)INDIQUE NÚMERO DA OPÇÃO''TT''
        (<RETURN>PARA TERMINAR)(WHT)''
1210 INPUT A$(TT)
1220 IF A$(TT)='' '' OR TT=51 THEN TT=TT-1:RETURN
1230 GOTO 1180
1240 REM ********************
1250 REM PERGUNTAS DE DISCRIMINAÇÃO
1260 PRINT''
(CLR)(PUR)''
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT''
(PUR)INDIQUE NÚMERO DA PERGUNTA ''DQ:PRINT''(<RETURN>PARA
     TERMINAR)(WHT)''
1340 INPUT E$(DQ)
1350 IF E$(DQ)='' '' OR DQ=51 THEN DQ=DQ-1:RETURN
1360 GOTO 1310
1370 REM ********************
1380 REM INICIALIZAÇÃO
1390 PRINT''
(CLR)'':POKE 53280,0:POKE 53281,0
1400 REM REDUZIR ARRAYS NA LINHA SEGUINTE?
1410 DIM A$(50),B(50,50),C(50),D(50),E$(50),F(50),G(50)
1420 X$='' ''
1430 PRINT''
(PUR)CARREGUE NUMA TECLA E EM <RETURN> SE''
1440 PRINT''DESEJA CONSULTAR BASE DE CONHECIMENTOS''
1450 PRINT''DEPOIS DA EXECUÇÃO; OU SÓ EM <RETURN>(WHT)''
```

```
1470 INPUT U$
1480 PRINT''
(CLR)''
1490 RETURN
1500 PRINT''
(PUR)---------------------------------------
----/WHT)''
1510 RETURN
2000 SID=54272
2010 FOR L1=0 TO 23
2020 POKE SID+L1,0
2030 NEXT L1
2040 POKE SID+24,7
2050 POKE SID+5,15
2060 POKE SID+6,55
2070 POKE SID+4,33
2080 FOR L1=1 TO 40 STEP 4
2090 POKE SID+1,L1
2100 FOR L2=1 TO 10
2110 NEXT L2,L1
2120 POKE SID+6,0
2130 RETURN
```

ESCALAS

```
10  REM ESCALAS
20  DIM X(50),Z(50)
30  PRINT''
(CLR)''
40  INPUT ''MAIOR VALOR'';A
50  INPUT ''MENOR VALOR'';B
60  A=A+.001
70  B=B+.001
80  C=C(A-B)/50
90  X(0)=B
100 FOR J=1 TO 50
110 X(J)=X(J-1)+C
120 Z(J)=J/50
130 PRINT Z(J),X(J)
140 NEXT J
150 DFF=(X(2)-X(1))/2
160 COUNT=0
170 COUNT=COUNT+1
180 PRINT''INDIQUE VALOR'';Q$
190 INPUT Q$
```

```
200 IF Q$='' ''THEN END
210 Q=VAL(Q$)
220 IF Q <B OR Q> A THEN 180
230 FOR J=1 TO 50
240 IF ABS(Q-X(J))<DFF THEN PRINT COUNT;''-'';Z(J)
250 NEXT J
260 GOTO 170
```

HASTE

```
10  REM HASTE
20  DIM A$(255),B$(255):F=0
30  PRINT''
(CLR)'':POKE 53280,0:POKE 53281,0
40  REM ********************
50  FLAG=0
60  D$='' '':INPUT''
(PUR)>  '';D$:PRINT''(WHT)''
70  IF D$='' '' THEN END
80  IF LEFT$(D$,12)=''?'' THEN 200
90  E=0
100 E=E+1
110 IF MID$(D$,E,1)=''*'' THEN 140
120 IF E<LEN(D$) THEN 100
130 PRINT ''INVALID ENTRY'':GOTO 50
140 IF FLAG=3 THEN RETURN
150 F=F+1:IF F=256 THEN END
160 A$(F)=LEFT$(D$,E-1)
170 B$(F)=MID$(D$,E+1)
180 GOTO 50
190 REM ********************
200 REM INTERROGAR
210 FLAG=4:TRUE=0
20 IF RIGHT$(D$,1)=''/'' THEN FLAG=3
230 FOR J=1 TO LEN(D$)-5
240 IF MID$(D$,J,5)='' AND '' THEN FLAG=15:TRUE=J
250 NEXT J
260 IF FLAG=5 THEN 410
270 IF LEFT$(D$,3)=''?/*'' THEN FLAG=1
280 IF LEFT$(D$,4)=''?/*/'' THEN FLAG=2
290 IF FLAG=3 THEN GOSUB 90:F$=MID$(D$,2,E-2)
300 IF FLAG=1 THEN F$=MID$(D$,4)
310 E=0
320 E=E+1
330 IF A$(E)='' '' AND FLAG=4 AND TRUE=0 THEN PRINT ''FALSE''
```

```
340 IF A$(E)='' '' THEN 520
350 IF FLAG=4 AND ''?''+A$(E)+''*''+B$(E)=D$ THEN PRINT
    «TRUE»:TRUE=1
360 IF FLAG=3 AND F$=A$(E) THEN PRINT B$(E)
370 IF FLAG=2 THEN PRINT A$(E);''*'';B$(E)
380 IF FLAG=1 AND F$=B$(E) THEN PRINT A$(E)
390 IF E<255 THEN 320
400 REM ********************
410 F$=MID$(D$,4,TRUE-4):G$=MID$(D$,TRUE+7)
420 E=0
430 E=E+1
440 IF A$(E)='' '' THEN 520
450 IF B$(E)=F$ THEN 470
460 IF E<255 THEN 430
470 H=0
480 H=H+1
490 IF B$(H)='' '' THEN 460
500 IF B$(H)=G$ AND A$(E)=A$(H) THEN PRINT A$(E)
510 IF H<255 THEN 480
520 PRINT TAB(5);''>END OF ANSWER<''
530 GOSUB 2000:GOTO 50
2000 SID=54272
2010 FOR L1=0 TO 23
2020 POKE SID+L1,0
2030 NEXT L1
2040 POKE SID+24,15
2050 POKE SID+5,15
2060 POKE SID+6,255
2070 POKE SID+4,17
2080 FOR L1=48 TO 220 STEP .7
2090 POKE SID+,L1
2100 NEXT L1
2110 FOR L1=28 TO 200
2120 POKE SID+1,L1
2130 NEXT L1
2140 FOR L1=200 TO 28 STEP -1
2150 POKE SID+1,L1
2160 NEXT L1
2170 POKE SID+1,0
2180 RETURN
```

EASLE

```
10 REM EASLE
20 PRINT''
```

```
(CLR)'':POKE 53280,0:POKE 53281,0
30 DIM X(40),Y(40),Z(40)
40 PRINT
50 A$='' '':INPUT ''>
(WHT)'';A$
60 IF A$='' '' THEN END
70 REM ********************
80 PRINT''
(PUR)'';:FOR J=1 TO 40
90 X(J)=0:Y(J)=0:Z(J)=0
100 NEXT J
110 REM ********************
120 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
130 FOR J=1 TO LEN(A$)
140 B$=MID$(A$,J,1)
150 IF B$=''('' THEN S=S+1:Z(S)=J:IF T=0 THEN CFIRST=J
160 IF B$='')'' THEN T=T+1:Y(T)=J:IF CSECND<>0 AND EDGE=0
    THEN EDGE=J-1
170 IF T=1 AND B$='')'' THEN CSECND=J
180 IF B$= '' '' THEN R=R+1:X(R)=J
190 NEXT J
200 IF S=T THEN 260:REM EQUILÍBRIO ( )
210 IF S<T THEN PRINT '' ->MISSING (''
220 IF S>T THEN PRINT '' ->MISSING )''
230 INPUT ''>
(WHT)'';B$:PRINT ''(PUR)'';
240 A$=A$+B$
250 GOTO 80
260 FLAG=0
270 IF LEFT$(A$,5)=''CAR ('' THEN FLAG=1
280 IF LEFT$(A$,5)=''CDR ('' THEN FLAG=2
290 IF LEFT$(A$,6)=''CONS ('' THEN FLAG=3
300 IF LEFT$(A$,6)=''ATOM ('' THEN FLAG=4
310 IF LEFT$(A$,4)=''EQ ('' THEN FLAG=5
320 IF LEFT$(A$,6)=''NULL ('' THEN FLAG=6
330 ON FLAG GOSUB 420,470,550,690,780,920
340 IF FLAG<>0 THEN GOSUB 360
350 GOTO 40
360 REM ** RESPOSTA **
370 GOSUB 4000:PRINT''   VALUE IS...''
380 IF B$<>''( )'' THEN PRINT''  '';B$
390 IF B$=''( )'' THEN PRINT''  NIL''
400 RETURN
410 REM ********************
420 REM ** CAR **
```

```
430 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z(2))
440 IF S>2 THEN B$=MID$(A$,CFIRST,CSECND-CFIRST+1)
450 RETURN
460 REM ********************
470 REM ** CDR **
480 GOSUB 420
490 LB=LEN(B$)+7
500 B$=''(''+MID$(A$,LB,EDGE-1)
510 IF RIGHT$(B$,2)=''))'' THEN B$=LEFT$(B$,LEN(B$)-1
520 IF MID$(B$,2,1)='' '' THEN B$=''(''+MID$(B$,3)
530 RETURN
540 REM ********************
550 REM ** CONS **
560 B$=MID$(A$,7,LEN(A$)-1)
570 J=0
580 IF LEFT$(B$,1)=''('' THEN J=1
590 J=J+1
600 IF MID$(B$,J,1)=''('' THEN 630
610 IF J<LEN(B$) THEN 590
620 B$=''  >CONS ERROR<'':RETURN
630 LB =LEN(B$)-1
640 B$=''(''+LEFT$(B$,J-1)+MID$(B$,J+1)
650 B$=LEFT$(B$,LB)
660 IF RIGHT$(B$,2)='' )'' THEN B$=LEFT$(B$,LEN(B$)-2)+'')''
670 RETURN
680 REM ********************
690 REM ** ATOM **
700 A$=MID$(A$,7,LEN(A$)-1)
710 J=0:B$=''NIL''
720 J=J+1
730 IF MID$(A$,J,1)='' '' OR MID$(A$,J,1)=''('' THEN RETURN
740 IF J<LEN(A$) THEN 720
750 B$=''T''
760 RETURN
770 REM ********************
780 REM ** EQ **
790 A$=MID$(A$,5)
800 A$=LEFT(A$,LEN(A$)-1)
810 J=0:B$=''NIL''
820 J=J+1
830 IF MID$(A$,J,1)='')'' THEN RETURN
840 IF MID$(A$,J,1)='' '' THEN 870
850 IF J<LEN(A$) THEN 820
860 RETURN
870 C$=LEFT$(A$,J-1)
```

```
880 A$=MID(A$,J+1)
890 IF C$=A$ THEN B$=''T''
900 RETURN
910 REM ********************
920 REM ** NULL **
930 B$=''NIL''
940 IF A$=''NULL ( )'' THEN PRINT''ILLEGAL - NULL NEEDS
    ARGUMENT'':B$='' ''
950 IF A$=''NULL (( ))'' THEN B$=''T''
960 RETURN
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP 3
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200 STEP 3
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -3
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

PROLOG-A

```
1   REM ********************
2   REM USE ''.'' EM VEZ DE '':''.POR
3   REM EXEMPLO, ''WHICH(X:SUM(1 1 2))''
4   REM PASSA A ''WHICH(X.SUM(1 1 2))''
5   REM ********************
10  REM PROLOG-A (SIMPLE FRONT-END)
20  REM TODAS AS ENTRADAS EM MAIÚSCULAS
30  GOTO 50
40  PRINT''
(BLU) NO (MORE) ANSWERS(WHT)'':RETURN
50  GOSUB 3270:REM INICIALIZAR
60  REM ********************
70  GOSUB 4000:PRINT
```

```
80  J$='' '':INPUT''
(PUR)&.(WHT)'';J$
90  IF J$='' '' THEN END
100 PRINT''
(PUR)'';:IF J$=''LIST ALL'' THEN GOSUB 1860:GOTO 70
110 IF LEFT$(J$,5)=''LIST '' THEN J$=MID$(J$,5)+'' '':GOSUB
    990:GOTO 70
120 IF RIGHT$(J$,1)<>'')'' THEN PRINT''1.'': INPUT
    M$:J$=J$+M$:GOTO 120
130 LJ=LEN(J$)
140 J$=LEFT$(J$,LJ-1)+'' '':REM TIRAR ) FINAL, SUBSTITUIR POR
    ESPAÇO
150 LJ=LEN(J$)
160 FLAG=0
170 IF LEFT$(J$,4)=''ADD(''THEN J$=MID$(J$,5):FLAG=1
180 RULEFLAG=0:PLUSFLAG=0:ARITHFLAG=0
190 FOR R=1 TO LEN(J$)
200 IF MID$(J$,R,4)='' IF '' THEN RULEFLAG=R:FLAG=6
210 IF MID$(J$,R,5)='' AND '' THEN PLUSFLAG=R
220 IF MID$(J$,R,4)=''SUM('' THEN ARITHFLAG=1
230 IF MID$(J$,R,6)=''TIMES('' THEN ARITHFLAG=2
240 IF MID$(J$,R,6)='' LESS '' THEN ARITHFLAG=3
250 IF MID$(J$,R,3)='' INT '' THEN ARITHFLAG=4
260 NEXT R
270 IF LEFT$(J$,3)3''IS('' THEN J$=MID$(J$,4):FLAG=2
280 IF LEFT$(J$,10)=WHICH(X . '' THEN J$=MID$(J$,11):FLAG=3
290 IF LEFT$(J$,16)=''WHICH((X Y) . X '' THEN
    J$=MID$(J$,17):FLAG=4
300 IF FLAG=0 THEN PRINT''SYNTAX ERROR'':GOTO 70
310 LJ=LEN(J$)
320 REM ENVIAR A SUBROTINAS RELEVANTES
330 IF PLUSFLAG<>0 THEN GOSUB 1950:GOTO 70:REM REGRA
    CONTENDO AND
340 IF RULEFLAG<>0 AND FLAG<>5 THEN GOSUB 1110:REM
    CODIFICAR REGRA
350 IF ARITHFLAG<>0 THEN GOSUB 2430:GOTO 70:REM ARITMÉTICA
360 IF RIGHT$(J$,3)='' X '' OR RIGHT$(J$,3)='' S '' THEN
    J$=LEFT$(J$,LJ-2)+'' ''
370 LJ=LEN(J)
380 IF FLAG=1 THEN GOSUB 440:REM ADD
390 IF FLAG=2 THEN GOSUB 520:REM IS
400 IF FLAG=3 THEN GOSUB 610:REM WHICH
410 IF FLAG=4 THEN GOSUB 830:REM WHICH2
420 GOTO 70
430 REM ********************
```

```
440 REM    ADD
450 K=0
460 K=K+1
470 IF Z$(K)='' ''THEN Z$(K)=J$:RETURN
480 IF K<1000 THEN 460
490 PRINT''MEMORY FULL''
500 RETURN
510 REM ********************
520 REM    IS
530 K=0
540 K=K+1
550 IF Z$(K)='' '' THEN 580
560 IF Z$(K)=J$ THEN PRINT''YES'':GOTO 590
570 IF K<1000 THEN 540
580 PRINT''NO''
590 RETURN
600 REM ********************
610 REM       WHICH
620 IF LEFT$(J$,1)=''X'' THEN 710
630 J$=LEFT$(J$,LJ-1)
640 K=0
650 K=K+1
660 IF Z$(K)='' '' THEN 690
670 IF J$=LEFT$(Z$(K),LEN(J$)) THEN PRINT
    RIGHT$(Z$(K),(LEN(Z$(K))-LEN(J$)))
680 IF K<1000 THEN 650
690 GOSUB 40
700 RETURN
710 REM * PERGUNTA COMEÇA POR X*
720 J$=MID$(J$,3,LEN(J$)-3)
730 LJ=LEN(J$)
740 K=0
750 K=K+1
760 IF Z$(K)='' '' THEN 800
770 Q$=MID$(Z$(K),LEN(Z$(K))-LJ,LJ)
780 IF Q$=J$ THEN PRINT LEFT$(Z$(K),LEN(Z$(K))-LJ+2))
790 IF K<1000 THEN 750
800 GOSUB
810 RETURN
820 REM ********************
830 REM  WHICH2
840 J$=LEFT$(J$,LJ-2)
850 LJ=LEN(J$)
860 K=0
870 K=K+1
```

```
880 IF Z$(K)='' '' THEN 960
890 LFLAG=0
900 FOR L=1 TO (LEN (Z$(K))-LJ)
910 IF MID$(Z$(K),L,LJ)=J$ THEN LFLAG=L
920 NEXT L
930 IF LFLAG=0 THEN 950
940 PRINT LEFT$(Z$(K),LFLAG-2);MID$(Z$(K),LFLAG+LJ))
950 IF K<1000 THEN 870
960 GOSUB 40
970 RETURN
980 REM ********************
990 REM   LIST
1000 K=0
1010 K=K+1
1020 IF Z$(K)='' '' THEN RETURN
1030 LFLAG=0
1040 FOR L=1 TO (LEN(Z$(K))-LEN(J$))
1050 IF MID$(Z$(K),L,LEN(J$))=J$)) THEN LFLAG=1
1060 NEXT L
1070 IF LFLAG=1 THEN PRINT Z$(K)
1080 IF K<1000 THEN 1010
1090 RETURN
1100 REM ********************
1110 REM   FORMAR REGRAS
1120 R=RULEFLAG
1130 E$=LEFT$(J$,R):F$=MID$(J$,R+4)
1140 IF LEFT$(E$,1)<>''X'' THEN PRINT''RULE ERROR'':GOTO 70
1150 REM DETECÇÃO DE ENTRADAS COMO X COME Y SE X É UM Y
1160 IF RIGHT$(F$,2)=''S'' THEN 1390
1170 PRINT TAB(18 ;''COMPILING RULE''
1180 FOR T=1 TO 100
1190 R$(T)='' ''
1200 NEXT T
1210 E$=MID$(E$,3):F$=MID$(F$,3)
1220 K=0:RR=0
1230 k=K+1
1240 IF Z$(JK)='' '' THEN 1300
1250 IF RIGHT$(Z$(K),LEN(F$))<>F$ THEN 1370
1260 RR=RR+1
1270 R$(RR)=LEFT$(Z$(K),(LEN(Z$(K))-LEN(F$)))+E$
1280 PRINT ''>  '';R$(RR)
1290 GOTO 1230
1300 IF RR=0 THEN RETURN
1310 RC=0
1320 RC=RC+1
```

```
1330 Z$(K)=R$(RC)
1340 IF K<1000 THEN K=K+1
1350 IF RC<RR THEN 1320
1360 RETURN
1370 IF K<1000 THEN 1230
1380 RETURN
1390 REM ** REGRA COM DUAS VARIÁVEIS *
1400 FOR T=1 TO 100
1410 R$(T)='' ''
1420 NEXT T
1430 K=0:RR=0
1440 IF K=1000 THEN RETURN
1450 K=K+1
1460 IF Z$(K)='' '' THEN 1770
1470 REM DIVIDIR EM TRÊS PALAVRAS
1480 Q$=Z$(K)
1490 J=0
1500 J=J+1
1510 IF MID$(Q$,J,1)=CHR$(32) THEN 1540
1520 IF J<LEN(Q$) THEN 1500
1530 PRINT''RULE COMPILING ERROR'':GOTO 70
1540 A$=LEFT$(Q$,J)
1550 Q$=MID$(Q$,J+1)
1560 J=0
1570 J=J+L
1580 IF MID$(Q$,J,1)=CHR$(32)THEN 1610
1590 IF J<LEN(Q$) THEN 1570
1600 PRINT''RULE COMPILING ERROR'':GOTO 70
1610 B$=LEFT$(Q$,J)
1620 Q$=MID$(Q$,J+1)
1630 J=0
1640 J=J+1
1650 IF MID$(Q$,J,1)=CHR$(32) THEN 1680
1660 IF J<LEN(Q$) THEN 1640
1670 PRINT''RULE COMPILING ERROR'':GOTO 70
1680 PRINT TAB(18);''COMPILING RULE''
1690 C$=LEFT$(Q$,J)
1700 M$=LID$(F$,3,LEN(B$))
1710 IF B$<>M$ THEN 1440
1720 RR=RR+1
1730 N$=MID$(E$,3,LEN(E$)-4)
1740 R$(RR)=A$+N$+C$
1750 PRINT''>    '';R$(RR)
1760 GOTO 1440
1770 IF RR=0 THEN RETURN
```

```
1780 M=0
1790 M=M+1
1800 IF M>RR THEN RETURN
1810 Z$(K)=R$(M)
1820 IF K=1000 THEN PRINT ''OUT OF MEMORY'':GOTO 70
1830 GOTO 1790
1850 REM ********************
1860 REM   LISTAR TUDO
1870 PRINT
1880 K=0
1890 K=K+1
1900 IF Z$(K)='' '' THEN RETURN
1910 PRINT Z$(K)
1920 IF K<1000 THEN 1890
1930 RETURN
1940 REM ********************
1950 REM FORMAR REGRAS COM ''AND'' DO SEGUINTE TIPO:
1960 REM (X COME Y SE X É UMA AVE E Y VEM EM CAIXAS)
1970 REM A DECLARAÇÃO X DEVE ESTAR ANTES DE Y EM TODA A LISTA
1980 REM DIVIDIR EM SECÇÕES
1990 J$=MID$(J$,2):REM TIRAR ''X''
2000 PRINT TAB(20);''COMPILING RULE''
2010 J=1
2020 J=J+1
2030 IF MID$(J$,J,1)=CHR$(32) THEN 2060
2040 IF J<LEN(J$) THEN 2020
2050 PRINT ''RULE COMPILING ERROR'':RETURN
2060 A$=LEFT$(J$,J):REM RELAÇÃO 1
2070 J$=MID$(J$,J+7):REM TIRAR ATÉ INÍCIO DA SEGUNDA RELAÇÃO
2080 J=1:COUNT=0
2090 J=J+1
2100 IF MID$(J$,J,1)=CHR$(32) THEN COUNT=COUNT+1
2110 IF COUNT=2 THEN 2140
2120 IF J<LEN(J$) THEN 2090
2130 PRINT''RULE COMPILING ERROR'':RETURN
2140 B$=LEFT$(J$,J):REM DECLARAÇÃO1
2150 C$=MID$(J$,J+6):REM DECLARAÇÃO2
2160 IF C$=CHR$(32) THEN PRINT''RULE COMPILING
     ERROR'':RETURN
2170 REM PERCORRER BASE DE DADOS
2180 FOR T=1 TO 200
2190 R$(T)='' ''
2200 NEXT T
2210 R1=0:R2=99
2220 K=0
```

```
2230 K=K+1
2240 IF Z$(K)='' '' THEN 2310
2250 IF R1=99 OR R2=200 THEN PRINT''MEMORY SHORTAGE'':GOTO 2310
2260 LB=LEN(B$)
2270 IF RIGHT$(Z$(K),LB)=B$ THEN
     R1=R1+1:R$(R1)=LEFT$(Z$(K),LEN(Z$(K))-LB)
2280 LC=LEN(C$)
2290 IF RIGHT$(Z$(K),LC)=C$ THEN R2=R2+1:R$(R2)=LEFT$(Z$(K),
     LEN(Z$(K),LEN(Z$(K))-LC)
2300 IF K<1000 THEN 2230
2310 IF R$(100)='' '' THEN PRINT''STATEMENT OF INPUT NOT IN
     DATABASE'':RETURN
2320 REM CODIFICAR REGRAS
2330 R1=0:R2=99
2340 R2=R2+1
2350 R1=R1+1
2360 IF R$(R1)>CHR$(32) AND R$(R2)>CHR$(32) THEN
     Z$(K)=R$(R1)+A$+R$(R2)+'' ''
2350 PRINT''>  '';Z$(K)
2380 K=K+1
2390 IF R$(R2+1)<>'' '' THEN 2340
2400 IF R$(R1+1)<>'' '' THEN 2350
2410 RETURN
2420 REM ********************
2430 REM   ARITMÉTICA
2440 LJ=LEN(J$)
2450 IF ARITHFLAG<3 THEN GOSUB 2490
2460 IF ARITHFLAG=3 THEN GOSUB 2890
2470 IF ARITHFLAG=4 THEN GOSUB 3080
2480 RETURN
2490 REM *** SUM, TIMES ***
2500 J$=MID$(J$,5,LJ-5)
2510 IF LEFT$(J$,2)=''S('' THEN J$=MID$(J$,3)
2520 LJ=LEN(J$)
2530 K=0
2540 K=K+1
2550 IF MID$(J$,K,1)=CHR$(32) THEN A$=LEFT$(J$,K-
     1):J$=MID$(J$,K+1): GOTO 2580
2560 IF K<LJ THEN 2540
2570 PRINT TAB(12);''ARITHMETIC ERROR'';RETURN
2580 LJ=LEN(J$)
2590 K=0
2600 K=K+1
2610 IF MID$(J$,K,1)=CHR$(32) THEN B$=LEFT$(J$,K-
     1)=J$=MID$(J$,K+1): GOTO 2460
```

```
2620 IF K<LJ THEN 2600
2630 PRINTTAB(12);''ARITHMETIC ERROR'':RETURN
2640 LJ=LEN(J$)
2650 K=0
2660 K=K+1
2670  IF MID$(J$,K,1)=CHR$(41)  THEN  C$=LEFT$(J$,K-1):GOTO
     2700
2680 IF K<LJ THEN 2660
2690 PRINTTAB(12);''ERROR (TOO MANY VARIABLES)'':RETURN
2700 AN=0:BN=0:CN=0
2710 IF ASC(A$)>58 THEN AN=1
2720 IF ASC(B$)>58 THEN BN=1
2730 IF ASC(C$)>58 THEN CN=4
2740 GUIDE=AN+BN+CN:IF GUIDE=3 OR GUIDE=5 OR GUIDE=6
     THEN 2960
2750 IF ARITHFLAG=2 THEN 2820:REM TIMES
2760 IF GUIDE>0 THEN 2790
2770 IF VAL(A$)+VAL(B$)=VAL(C$) THEN PRINT''YES'':RETURN
2780 PRINT''NO'':RETURN
2790 IF GUIDE=1 THEN PRINTVAL(C$)-VAL(B$):GOSUB 40:RETURN
2800 IF GUIDE=2 THEN PRINTVAL(C$)-VAL(A$):GOSUB 40:RETURN
2810 PRINT VAL(A$)+VAL(B$):GOSUB 40:RETURN
2820 REM ** TIMES **
2830 IF GUIDE>0 THEN 2860
2840 IF VAL(A$)*VAL(B$)=VAL(C$) THEN PRINT''YES'':RETURN
2850 PRINT''NO'':RETURN
2860 IF GUIDE=1 THEN PRINTVAL(C$)/VAL(B$);GOSUB 40:RETURN
2870 IF GUIDE=2 THEN PRINTVAL(C$)/VAL(A$):GOSUB 40:RETURN
2880 PRINT VAL(A$)*VAL(B$):GOSUB 40:RETURN
2890 REM *************** LESS ***************
2900 NF=0
2910 IF ASC(J$)<58 THEN NF=1:REM NUMEROS
2920 COUNT=0
2930 K=0
2940 K=K+1
2950 IF MID$(J$,K,1)=CHR$(32) THEN COUNT=COUNT+1
2960 IF COUNT=2 THEN 3000
2970 IF K<LEN(J$) THEN 2940
2980 PRINT TAB(20)''COMPARISON ERROR''
2990 RETURN
3000 B$=MID$(J$,K+1)
3010 A$=LEFT$(J$,K-6)
3020 IF NF=1 THEN 3050
3030 IF A$<B$ THEN PRINT ''YES'':RETURN
3040 PRINT''NO'':RETURN
```

```
3050 REM * NÚMEROS *
3060 IF VAL(A$)<VAL(B$) THEN PRINT''YES'':RETURN
3070 PRINT''NO'':RETURN
3080 REM ********* INT **********
3090 IF RIGHT$(J$,2)=''X'' THEN 3190
3100 K=0
3110 K=K+1
3120 IF MID$(J$,K,1)=CHR$(32) THEN 3160
3130 IF K<LEN(J$) THEN 3110
3140 PRINT TAB(20);''ARITHMETIC ERROR''
3150 RETURN
3160 A=VAL(LEFT$(J$,K-1))
3170 IF INT(A)=A THEN PRINT''YES'':RETURN
3180 PRINT''NO'':RETURN
3190 K=0
3200 K=K+1
3210 IF MID$(J$,K,1)=CHR$(32) THEN 3240
3220 IF K<LEN(J$) THEN 3200
3230 PRINT TAB(20);''ARITHMETIC ERROR'':RETURN
3240 PRINT INT (VAL(LEFT$(J$,K-1)))
3250 RETURN
3260 REM ********************
3270 REM   INICIALIZAR
3280 PRINT''
(CLR)'':POKE 53280,0;POKE 53281,0
3290 DIMZ$(1000),R$(200)
3300 RETURN
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP 3
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200 STEP 3
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -3
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

SSLISP

```
10  REM S.S. LISP
20  GOTO 3450
30  REM ********************
40  A$=MID$(A$,E):A$=LEFT$(A$,LEN(A$)-1
50  RETURN
60  REM ********************
70  PRINT ''
(PUR)''
80  NN=0
90  A$='' '':INPUT'':
(WHT)'';A$
100 IF A$='' '' THEN END:REM CARREGAR EM <RETURN> PARA
TERMINAR
110 REM ********************
120 PRINT ''
(PUR)'';:FOR J=1 TO 12
130 X(J)=0:Y(J)=0:Z(J)=0
140 NEXT J
150 REM ********************
160 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
170 FOR J=1 TO LEN(A$)
180 B$=MID$(A$,J,1)
190 IF B$=''(''THEN S=S+1:Z(S)=J:IF T=0 THEN CFIRST=J
200 IF B$='')'' THEN T=T+1:Y(T)=J:IF CSECND<>0 AND EDGE=0
    THEN EDGE=J-1
210 IF T=1 AND B$='')'' THEN CSECND=J
220 IF B$='' '' THEN R=R+1:X(R)=J
230 NEXT J
240 IF S=T THEN 300:REM EQUILÍBRIO ( )
250 IF S<T THEN PRINT '' ->MISSING (''
260 IF S>T THEN PRINT '' ->MISSING )''
270 INPUT ''+ '';B$
280 A$=A$+B$
290 GOTO 120
300 IF NWDS=0 OR NN=1 THEN 370
310 M$=LEFT$(A$,X(1)-1)
320 FOR J=1 TO NWDS
330 IF M$=N$(J) THEN A$=O$(J)+MID$(A$,LEN(N$(J))+1)
340 NEXT J
350 NN=1
360 GOTO 120
370 FLAG=0:B$=''NIL''
380 IF LEFT$(A$,5)=''CAR ('' THEN FLAG=1
```

```
390 IF LEFT$(A$,5)=''CDR ('' THEN FLAG=2
400 IF LEFT$(A$,6)=''CONS ('' THEN FLAG=3
410 IF LEFT$(A$,6)=''ATOM'' ('' THEN FLAG=4
420 IF LEFT$(A$,4)=''EQ ('' THEN FLAG=5
430 IF LEFT$(A$,6)=''NUL ('' THEN FLAG=6
440 IF LEFT$(A$,8)=''MEMBER ('' THEN FLAG=7
450 IF LEFT$(A$,6)=''MEMQ (''THEN FLAG=8
460 IF LEFT$(A$,8)=''APPEND (''THEN FLAG=9
470 IF LEFT$(A$,9)=''REVERSE ('' THEN FLAG=10
480 IF LEFT$(A$,7)=''EQUAL ('' THEN FLAG=11
490 IF LEFT$(A$,6)=''LIST ('' THEN FLAG=12
500 IF LEFT$(A$,8)=''DEFINE ('' THEN FLAG=13
510 IF LEFT$(A$,6)=''ADD1 ('' THEN FLAG=14
520 IF LEFT$(A$,7)=''SUB1 ('' THEN FLAG=15
530 IF LEFT$(A$,7)=''ZEROP ('' THEN FLAG=16
540 IF LEFT$(A$,12)=''DIFFERENCE ('' THEN FLAG=17
550 IF LEFT$(A$,6)=''EXPT ('' THEN FLAG=18
560 IF LEFT$(A$,5)=''MAX ('' THEN FLAG=19
570 IF LEFT$(A$,5)=''MIN ('' THEN FLAG=20
580 IF LEFT$(A$,6)=''PLUS ('' THEN FLAG=21
590 IF LEFT$(A$,7)=''MINUS ('' THEN FLAG=22
600 IF LEFT$(A$,10)=''QUOTIENT ('' THEN FLAG=23
610 IF LEFT$(A$,7)=''RECIP ('' THEN FLAG=24
620 IF LEFT$(A$,11)=''REMAINDER ('' THEN FLAG=25
630 IF LEFT$(A$,7)=''TIMES ('' THEN FLAG=26
640 IF LEFT$(A$,10)=''GREATERP ('' THEN FLAG=27
650 IF LEFT$(A$,7)=''LESSP ('' THEN FLAG=28
660 IF LEFT$(A$,8)=''MINUSP ('' THEN FLAG=29
670 IF LEFT$(A$,9)=''NUMBERP ('' THEN FLAG=30
680 IF LEFT$(A$,6)=''ONEP ('' THEN FLAG=31
690 IF FLAG>13 THEN 720
700 ON FLAG GOSUB 840,890,970,1110,1200,1330,1380,1510,1700,
    1760,2000,2130,3300
710 GOTO 760
720 IF FLAG>24 THEN 750
730 ON (FLAG-13) GOSUB 2180,2230,2280,2350,2560,2600,2790,
    2820,2870,2920,2960
740 GOTO 760
750 ON (FLAG-24) GOSUB 3030,3070,3120,3160,3200,3250,2280
760 IF FLAG<>0 THEN GOSUB 780
770 GOTO 70
780 REM ** RESPOSTA **
790 GOSUB 4000: PRINT''   VALUE IS...
800 IF B$<>''( )'' THEN PRINT''   '';B$
810 IF B$=''( )'' THEN PRINT''   NIL''
```

```
820 RETURN
830 REM ********************
840 REM ** CAR **
850 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z(2))
860 IF S> 2 THEN B$=MID$(A$,CFIRST,CSECND-CFIRST+1)
870 RETURN
880 REM ********************
890 REM ** CDR **
900 GOSUB 840
910 LB=LEN(B$)+7
920 B$=''(''+MID$(A$,LB,EDGE-1)
930 IF RIGHT$(B$,2)=''))'' THEN B$=LEFT$(B$,LEN(B$)-1)
940 IF MID$(B$,2,1)='' ''THEN B$=''(''+MID$(B$,3)
950 RETURN
960 REM ********************
970 REM ** CONS **
980 B$=MID$(A$,7,LEN(A$)-1)
990 J=0
1000 IF LEFT$(B$,1)=''('' THEN J=1
1010 J=J+1
1020 IF MID$(B$,J,1)=''('' THEN 1050
1030 IF J<LEN (B$) THEN 1010
1040 B$=''   >CONS ERROR<'':RETURN
1050 LB=LEN(B$)-1
1060 B$=''(''+LEFT$(B$,j-1)+MID$(B$,J+1)
1070 B$=LEFT$(B$-LB)
1080 IF RIGHT$(B$,2='' )'' THEN B$=LEFT$(B$,LEN(B$)-2)+'')''
1090 RETURN
1100 ********************
1110 REM ** ATOM **
1120 A$=MID$(A$,7,LEN(A$)-1)
1130 J=0:B$=''NIL''
1140 J=J+1
1150 IF MID$(A$,J,1)='' '' OR MID$(A$,J,1)=''('' THEN RETURN
1160 IF J<LEN(A$) THEN 1140
1170 B$=''T''
1180 RETURN
1190 REM ********************
1200 REM ** EQ **
1210 E=5:GOSUB 40
1220 J=0
1230 J=J+1
1240 IF MID$(A$,J,1)='')'' THEN RETURN
1250 IF MID$(A$,J,1)='' ''THEN 1280
1260 IF J<LEN(A$) THEN 1230
```

```
1270 RETURN
1280 C$=LEFT$(A$,J-1)
1290 A$=MID$(A$,J+1)
1300 IF C$=A$ THEN B$=''T''
1310 RETURN
1320 REM ********************
1330 REM ** NULL **
1340 IF A$=''NULL ( )'' THEN B$='' * ILLEGAL - NULL NEEDS AR-
     GUMENT *''
1350 XX$=''NULL (( ))'':IF A$=XX$ THEN B$=''T''
1360 RETURN
1370 REM ********************
1380 REM ** MEMBER **
1390 C$=MID$(A$,9)
1400 J=1
1410 J=J+1
1420 IF MID$(C$,J,1)='')'' OR MID$(C$,J,1)=''(''THEN
     D$=LEFT$(C$,J) :GOTO 1450
1430 IF J<LEN(C$) THEN 1410
1440 RETURN
1450 J=LEN(D$)
1460 J=J+1
1470 IF MID$(C$,J,LEN(D$))=D$ THEN C$=LEFT$(C$,LEN(C$)-
     1):GOTO 1630
1480 IF J<LEN(C$) THEN 1460
1490 RETURN
1500 REM ********************
1510 REM ** MEMQ **
1520 C$=MID$(A$,7)
1530 J=0
1540 J=J+1
1550 XX$='' '':IF MID$(C$,J,1)=XX$ THEN 1580
1560 IF J<LEN(A$) THEN 1540
1570 RETURN
1580 D$=LEFT$(C$,J)
1590 C$=MID(C$,J+2)
1600 C$=LEFT$(C$,LEN(C$)-2)+'' ''
1610 J=0
1620 J=J+1
1630 IF MID$(C$,J,LEN(D$))=D$ THEN B$=''(''+MID$(C$,J):GOTO 1660
1640 IF J<LEN(C$) THEN 1620
1650 RETURN
1660 B$=LEFT$(B$,LEN(B$)-1)+'')''
1670 IF RIGHT$(B$,3)=''))'' THEN B$=LEFT$(B$,LEN(B$)-
     1):GOTO 1670
```

```
1680 RETURN
1690 REM ********************
1700 REM ** APPEND **
1710 B$=MID$(A$,9)
1720 B$=LEFT$(B$,Y(1)-9+'' ''+MID$(B$,Z(3)-7)
1730 B$=LEFT$(B$,LEN(B$)-1)
1740 RETURN
1750 REM *******************
1760 REM ** REVERSE **
1770 B$='' ''
1780 A$=MID$(A$,11):A$=LEFT$(A$,LEN(A$)-2)
1790 CT=1
1800 J=0
1810 J=J+1:IF J>LEN(A$) THEN 1920
1820 XX$=MID$(A$,J,1):IF XX$='' '' THEN GOTO 1580
1830 IF XX$=''('' THEN 1860
1840 GOTO 1810
1850 CT=CT+1:G$(CT)=LEFT$(A$,J-1):A$=MID$(A$,J+1):GOTO
     1800
1860 J=J+1;IF MID$(A$,J,2)=''))'' THEN 1980
1870 IF MID$(A$,J,1)='')'' THEN 1910
1880 IF J=LEN(A$) THEN 1900
1890 GOTO 1860
1900 CT=CT+1:G$(CT)=A$+'')'':GOTO 1930
1910 CT= CT+1:G$(CT)=LEFT$(A$,J):A$=MID$(A$,J+=1):GOTO 1800
1920 CT=CT+1:G$(CT)=A$
1930 FOR M=CT TO 1 STEP -1
1940 B$=B$+G$(M):IF M>1 THEN B$=B$+'' ''
1950 NEXT M
1960 B$=''(''+B$+'')''
1970 RETURN
1980 CT=CT+1:G$(CT)=LEFT$(A$,J+1):A$=MID$(A$,J+2):GOTO 1800
1990 REM *******************
2000 REM ** EQUAL **
2010 E=8:GOSUB 40
2020 M=ASC(A$):IF M>47 AND M<58 THEN 2370
2030 J=0
2040 J =J+1
2050 IF MID$(A$,J,2)='') '' THEN J=J+1:GOTO 1280
2060 IF MID$(A$,J,3)=''))'' THEN 2100
2070 IF MID$(A$,J,2)=''))'' THEN 2110
2080 IF J<LEN(A$) THEN 2040
2090 RETURN
2100 C$=LEFT$(A$,J+2):A$=MID$(A$,J+4):GOTO 1300
2110 C$=LEFT$(A$,J+1):A$=MID$(A$,J+3);GOTO 1300
```

```
2120 REM ********************
2130 REM ** LIST **
2140 E=7:GOSUB 40
2150 B$=''(''+A$+='')''
2160 RETURN
2170 REM ********************
2180 REM ** ADD1 **
2190 E=7:GOSUB 40
2200 B$=STR$(VAL(A$)+1)
2210 RETURN
2220 REM ********************
2230 REM ** SUB1 **
2240 E=7:GOSUB 40
2250 B$=STR$(VAL(A$)-1)
2260 RETURN
2270 REM *******************
2280 REM ** ZEROP **
2290 IF FLAG=16 THEN E=8
2300 IF FLAG=31 THEN E=7
2310 GOSUB 40
2320 IF (A$=''0'' AND FLAG=16) OR (A$=''1'' AND FLAG=31
     THEN B$=''T''
2330 RETURN
2340 REM ********************
2350 REM ** DOIS ARGUMENTOS **
2360 E=13:GOSUB 40
2370 J=0
2380 J=J+1
2390 IF MID$(A$,J,1)='' '' THEN 2420
2400 IF J<LEN(A$) THEN 2380
2410 B$='' * ERROR - ONLY ONE ARGUMENT *'':RETURN
2420 P=VAL(LEFT$(A$,J-1))
2430 Q=VAL(MID$(A$,J+1))
2440 IF FLAG=17 THEN B$=STR$(P-Q):RETURN
2450 IF FLAG=23 OR FLAG=25 THEN B=P/Q
2460 IF FLAG=25 THEN B=INT(.5+Q*(B-INT(B))*1000)/1000
2470 IF FLAG=18 THEN B=P ^Q
2480 IF FLAG=11 AND P=Q THEN B$=''T''
2490 IF FLAG=27 AND P>Q THEN B$=''T''
2500 IF FLAG=28 AND P<Q THEN B$=''T''
2510 IF FLAG=32 THEN B=P-Q
2520 IF FLAG=11 OR FLAG>26 THEN RETURN
2530 B$=STR$(B)
2540 RETURN
2550 REM *******************
```

```
2560 REM ** EXPT **
2570 E=7:GOSUB 40
2580 GOTO 2370
2590 REM ********************
2600 REM ** MAX (MIN PLUS TIMES) **
2610 F$=LEFT$(A$,3):A$=MID$(A$,6)
2620 CT=0:FLAG=0
2630 IF F$=''TIMES'' THEN CT=1
2640 J=0
2650 J=J+1
2660 IF MID$(A$,J,1)='' '' THEN 2690
2670 IF J<LEN(A$) THEN 2650
2680 IF J=LEN (A$) THEN FLAG=1
2690 P=VAL(LEFT$(A$,J-1)):IF FLAG=0 THEN A$=MID$(A$,J+1)
2700 IF F$<>''PLUS'' AND CT=0 THEN CT=P
2710 IF F$=''MAX'' AND P>CT THEN CT=P
2720 IF F$=''MIN'' AND P<CT THEN CT=P
2730 IF F$=''PLUS'' THEN CT=CT+P
2740 IF F$=''TIMES'' THEN CT=CT*P
2750 IF FLAG=0 THEN 2640
2760 B$=STR$(CT)
2770 RETURN
2780 REM ********************
2790 REM ** MIN **
2800 GOTO 2610
2810 REM ********************
2820 REM ** PLUS **
2830 F$=''PLUS''
2840 A$=MID$(A$,7)
2850 GOTO 2620
2860 REM ********************
2870 REM ** MINUS **
2880 E=8:GOSUB 40
2890 B$=STR$(-VAL(A$))
2900 RETURN
2910 REM ********************
2920 REM ** QUOCIENTE **
2930 E=11:GOSUB 40
2940 GOTO 2370
2950 REM ********************
2960 REM ** RECIP **
2970 E=8:GOSUB 40
2980 IF B$=''0'' THEN B$='' DIVISION BY ZERO ILLEGAL'':RETURN
2990 B=1/(VAL(A$))
3000 B$=STR$(B)
```

```
3010 RETURN
3020 REM ********************
3030 REM ** REMAINDER **
3040 E=12:GOSUB 40
3050 GOTO 2370
3060 REM ********************
3070 REM ** TIMES **
3080 F$=''TIMES''
3090 A$=MID$(A$,8)
3100 GOTO 2620
3110 REM ********************
3120 REM ** GREATERP **
3130 E=11:GOSUB 40
3140 GOTO 2370
3150 REM ********************
3160 REM ** LESSP **
3170 E=8:GOSUB 40
3180 GOTO 2370
3190 REM ********************
3200 REM **MINUSP **
3210 E=9:GOSUB 40
3220 IF VAL(A$)<0 THEN B$=''T''
3230 RETURN
3240 REM ********************
3250 REM ** NUMBERP **
3260 A$=MID$(A$,10)
3270 IF ASC(A$)>44 AND ASC(A$)<58 THEN B$=''T''
3280 RETURN
3290 REM ********************
3300 REM ** DEFINE **
3310 A$=MID$(A$,9)
3320 F$=LEFT$(A$,X(2)-9)
3330 G$=MID$(A$,X(4)-6)
3340 J=0
3350 J=J+1
3360 IF MID$(G$,J,1)='' '' THEN 3390
3370 IF J<LEN(G$) THEN 3350
3380 B$='' DEFINE ERROR'':RETURN
3390 G$=LEFT$(G$,J-1)
3400 NWDS=NWDS+1
3410 O$(NWDS)=G$:N$(NWDS)=F$
3420 B$=F$
3430 RETURN
3440 REM ********************
3450 REM INICIALIZAR
```

```
3460 PRINT''
(CLR)'':POKE 53280,0:POKE 53281,0
3470 DIM G$(20),0$(20),N$(20),X(12),Y(12),Z(12)
3480 NWDS=0:REM CONTAGEM DE PALAVRAS NOVAS
3490 GOTO 70
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP 3
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200 STEP 3
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -3
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

MECÂNICO

```
10  REM MECÂNICO
12  POKE 23658,8
14  POKE 23609,940
16  POKE 23692,255
20  BORDER 1: PAPER 1: INK 7:CLS
30  PRINT INK 5;''TEM PORTANTO PROBLEMAS COM O SEU CARRO''
32  BEEP .4,-4: BEEP .6,-12
34  PAUSE 50
40  PRINT INK 6;''VAMOS TENTAR DESCOBRIR A CAUSA''
42  BEEP .6,4: BEEP .4,12
70  PRINT
80  PRINT ''A - O MOTOR DE ARRANQUE NÃO FUNCIONA''
90  PRINT ''B - FUNCIONA, MAS O MOTOR DO CARRO NÃO''
100 PRINT ''C - ARRANCA, MAS PÁRA EM SEGUIDA''
110 PRINT ''D - ARRANCA, MAS VAI ABAIXO MUITAS VEZES''
120 PRINT ''E - ARRANCA, MAS É IRREGULAR EM PONTO MORTO''
130 IF INKEY$<>'' '' THEN GO TO 130
140 LET A$=INKEY$
150 IF A$<''A'' OR A$>''E''THEN GO TO 140
160 GO SUB 1560
170 IF A$=''A'' THEN GO TO 200
172 IF A$=''B'' THEN GO TO 580
174 IF A$=''C'' THEN GO TO 1200
176 IF A$=''D'' THEN GO TO 1230
178 IF A$=''E'' THEN GO TO 1350
180 STOP
190 REM ***********************
200 REM NÃO FUNCIONA
210 PRINT ''COMEÇAMOS POR VERIFICAR A BATERIA''
220 PRINT ''ACENDA AS LUZES''
230 PRINT TAB 6; INK 6; ''ESTÃO FRACAS?''
240 GO SUB 1520
```

```
250 IF A$=''N'' THEN GO TO 350: REM BATERIA OK
260 PRINT INK 6;''OS CABOS DA BATERIA ESTÃO SOLTOS OU
    CORROÍDOS?''
270 GO SUB 1520
280 IF A$=''S'' THEN PRINT ''APERTE-OS E LIMPE-OS''
290 GO SUB 1580
300 PRINT INK 6;''A CORREIA DA VENTOINHA ESTÁ POUCO
    APERTADA?''
310 GO SUB 1520
320 IF A$=''S'' THEN PRINT TAB 4;''APERTE-A MELHOR''
330 GO SUB 1580
340 PRINT ''CABOS LIGADOS A OUTRA BATERIA OU UM EMPURRÃO
    DEVEM BASTAR PARA ARRANCAR O CARRO'' : STOP
350 PRINT INK 6;''A EXTREMIDADE DO CABO DA BATERIA JUNTO AO
    MOTOR DE ARRANQUE ESTÁ DESAPERTADA OU CORROÍDA?''
370 GO SUB 1520
380 IF A$=''S'' THEN PRINT ''APERTE AS LIGAÇÕES E LIMPE-AS''
390 GO SUB 1580
400 PRINT ''COLOQUE UMA PEÇA METÁLICA ENTRE''
410 PRINT ''OS TERMINAIS DO SOLENÓIDE'',INK 6;''O MOTOR DE
    ARRANQUE FUNCIONA?''
420 GO SUB 1520
430 IF A$=''S'' THEN PRINT ''O BOTÃO DE IGNIÇÃO ESTÁ
    PROVAVELMENTE DEFEITUOSO''
440 GO SUB 1580
450 IF A$=''S'' THEN PRINT TAB 4; ''DEVE SER SUBSTITUÍDO'':
    STOP
460 PRINT INK 6;''O MOTOR PARECE QUERER ARRANCAR?''
470 GO SUB 1520
480 IF A$=''S'' THEN PRINT ''O MOTOR DE ARRANQUE PODE ESTAR
    PRESO'': GO TO 520
490 PRINT ''COMO NADA ACONTECEU, O SOLENÓIDE''
500 PRINT ''É PROVAVELMENTE RESPONSÁVEL. UM''
510 PRINT ''EMPURRÃO PODERÁ ARRANCAR O CARRO'': STOP
520 REM MOTOR DE ARRANQUE PRESO
530 PRINT ''DESLIGUE A IGNIÇÃO, E PONHA O''
540 PRINT ''CARRO EM QUARTA VELOCIDADE.''
550 PRINT ''EMPURRE-O CERCA DE 3 CENTÍMETROS PARA LIBERTAR O
    MOTOR DE ARRANQUE'': STOP
560 RETURN
570 REM ************************
580 REM RODA MAS NÃO ARRANCA
590 PRINT ''VERIFIQUE SE HÁ HUMIDADE NAS EXTREMIDADES DOS
    CABOS''
600 PRINT INK 6;''PARECE-LHE HAVER HUMIDADE?''
```

```
610 GO SUB 1520
620 IF A$=''S'' THEN PRINT ''USE UM PULVERIZADOR CONTRA A
    HUMIDADE'': GO SUB 1580
630 PRINT INK 6;''ENCONTRA POEIRA NA PARTE''
640 PRINT INK 6''ISOLANTE DO ENROLAMENTO OU DA''
650 PRINT INK 6''TAMPA DO DISTRIBUIDOR?
660 GO SUB 1520
670 IF A$=''S'' THEN PRINT ''LIMPE O INTERIOR E A TAMPA DO''
680 IF A$=''S'' THEN PRINT ''DISTRIBUIDOR'': GO SUB 1580
690 PRINT ''VERIFIQUE SE TODOS OS CABOS ESTÃO BEM APERTADOS E
    SECOS ANTES DE CONTINUAR''
700 GO SUB 1580
710 PRINT ''SE O MOTOR AINDA NÃO ARRANCA, É CHEGADO O MOMENTO
    DE VERIFICAR AS VELAS:''
720 GO SUB 1580
730 PRINT INK 6;''A FAÍSCA NA EXTREMIDADE DO CABO DA VELA
    SALTA MUITO?''
740 GO SUB 1520
750 IF A$=''S'' THEN PRINT ''AS VELAS NÃO ESTÃO EM BOM
    ESTADO'':GO SUB 1580
760 IF A$=''N'' THEN GO TO 830
770 PRINT INK 6;''AS VELAS ESTÃO GORDUROSAS?''
780 GO SUB 1520
790 IF A$=''S'' THEN PRINT ''NÃO É POSSÍVEL FAZER UMA
    REPARAÇÃO DE EMERGÊNCIA ENQUANTO AS''
800 IF A$=''S'' THEN PRINT ''VELAS SE ENCONTRAM NESSE
    ESTADO.'':GO SUB 1580: GO TO 770
810 IF A$=''N'' THEN PRINT ''SE SE TRATA DE UMA EMERGÊNCIA,''
820 IF A$=''N'' THEN PRINT ''EXPERIMENTE DIMINUIR A FOLGA
    PARA METADE DO NORMAL'': GO SUB 1580: GO TO 910
830 PRINT INK 6;''OBSERVA FAÍSCA?''
840 GO SUB 1520
850 IF A$=''S'' THEN GO TO 770
860 GO SUB 870: STOP
870 PRINT ''VERIFIQUE SE HÁ RACHAS NO ROTOR,''
880 PRINT ''ENROLAMENTO E TAMPA DO DISTRI-''
890 PRINT ''BUIDOR. SE NÃO HOUVER, O PROBLEMA DEVE RESIDIR
    NOS PLATINADOS OU NO CONDENSADOR''
900 PRINT ''PODERÁ SER NECESSÁRIO IR A UMA OFICINA'': RETURN
910 PRINT INK 6;''TEM GASOLINA NO DEPÓSITO?''
920 GO SUB 1520
930 IF A$=''N'' THEN PRINT ''ENCHA O DEPÓSITO E TENTE DE
    NOVO'': GO SUB 1580
940 PRINT INK 6; ''OS TUBOS DE COMBUSTÍVEL E AR ESTÃO BEM
    PRESOS?''
```

```
950 GO SUB 1520
960 IF A$=''N'' THEN PRINT ''FIXE-OS BEM E TENTE DE NOVO'': GO
    SUB 1580
970 PRINT ''REMOVA O FILTRO DE AR''
980 GO SUB 1580
990 PRINT INK 6;''PARECE SECO''
1000 GO SUB 1520
1010 IF A$=''N'' THEN GO TO 1080
1020 PRINT ''RODE UM POUCO O MOTOR TAPANDO''
1030 PRINT ''COM A MÃO A ENTRADA DE AR'': GO SUB 1580
1040 PRINT INK 6;''A MÃO FICOU HÚMIDA DE GASOLINA?''
1050 GO SUB 1520
1060 IF A$=''N'' THEN PRINT ''DESAPERTE TAMPA DA GASOLINA NO''
1070 PRINT ''CASO DE A ENTRADA DE AR ESTAR VEDADA. A BOMBA
     DE GASOLINA PODE NÃO ESTAR A TRABALHAR''. GO SUB 900:
     STOP
1080 PRINT INK 6: ''TENTOU LIGAR O MOTOR MUITAS VEZES ''
1090 PRINT ''NOS ÚLTIMOS MINUTOS?''
1100 GO SUB 1520
1110 IF A$=''N'' THEN GO TO 1150
1120 PRINT ''ESPERE CERCA DE UM MINUTO E CAR-''
1130 PRINT ''REGUE NO ACELERADOR ATÉ AO FUNDO, SEM O BOMBEAR.
     DEVE CONSEGUIR ARRANCAR'': STOP
1150 PRINT ''O MOTOR PODE ESTAR AFOGADO''
1160 PRINT ''E A VÁLVULA FIXA NA POSIÇÃO ABERTA.'': GO SUB
     1580
1170 PRINT ''BATA NO LADO DO CARBURADOR E''
1180 PRINT ''TENTE DE NOVO ARRANCAR'': STOP
1190 REM ***********************
1200 REM ARRANCA, VAI ABAIXO
1210 PRINT ''ISTO SUGERE UMA RESISTÊNCIA DEFEITUOSA, QUE DEVE
     SER SUBSTITUIDA'': STOP
1220 REM ***********************
1230 REM FUNCIONA, FALHA MUITO
1240 PRINT ''O PROBLEMA É CAUSADO POR''
1250 PRINT ''CABOS EM CURTO-CIRCUITO (OU MAL FIXOS);''
1260 PRINT ''FAÍSCA FRACA; OU''
1270 PRINT ''UMA FALHA NO SISTEMA DE COMBUSTÍVEL''
1280 PRINT ''VERIFIQUE PRIMEIRO OS CABOS.'': GO SUB 1580
1290 PRINT ''SE NÃO ESTIVEREM BEM, REPARE-OS. SE ESTIVEREM,
     PODERÁ TRATAR-SE DAS VELAS.''
1300 GO SUB 870
1310 PRINT ''PODE VERIFICAR AINDA AS VELAS''
1320 GO SUB 1580
1330 GO TO 1360
```

```
1340 REM ***********************
1350 REM FUNCIONA, IRREGULAR EM PONTO MORTO
1360 PRINT ''PODE ACONTECER QUE UMA OU MAIS''
1370 PRINT ''VELAS ESTEJAM DEFEITUOSAS.'': GO SUB 1580
1380 PRINT ''DESLIGUE-AS UMA DE CADA VEZ.''
1390 PRINT ''AS QUE NÃO PROVOCAREM PARAGEM''
1400 PRINT ''DO MOTOR ESTÃO DEFEITUOSAS. VERIFIQUE-AS E VOLTE
     AO SISTEMA.'': GO SUB 1580
1420 PRINT INK 6;''DESCOBRIU VELAS DEFEITUOSAS?''
1430 GO SUB 1520
1440 IF A$=''S'' THEN PRINT''SUBSTITUA TODAS AS VELAS, SE''
1450 IF A$=''S'' THEN PRINT''PUDER, OU SÓ AS DEFEITUOSAS'':
     GO TO 1580
1460 PRINT ''A CAUSA MAIS COMUM DE UM MAU''
1470 PRINT ''PONTO MORTO, SE AS VELAS ESTÃO''
1480 PRINT ''BOAS, É UMA MISTURA DO COMBUSTÍ-''
1490 PRINT ''VEL DEMASIADO RICA. DEVE PORTANTO AJUSTÁ-LA'':
     STOP
1500 PRINT ''AJUSTE O JIGLER NO CARBURADOR'': STOP
1510 REM ***********************
1520 PRINT TAB 10; INVERSE 1; INK 6;''(S - SIM, N - NÃO)?''
1530 IF INKEY$<>'' '' THEN GO TO 1530
1540 LET A$=INKEY$
1550 IF A$<>''S'' AND A$<>''N'' THEN GO TO 1540
1560 PRINT TAB 12;''>  OK   '';A$;''<''
1570 BEEP .03,12: BEEP .03,9: BEEP .03,15: BEEP .03,18
1580 PAUSE 100
1590 PRINT
1595 POKE 23692,255
1600 RETURN
```

MEDICI

```
10  REM MEDICI
15  POKE 23658,8
20  BORDER 1: PAPER 1: INK 7: CLS
30  GO SUB 1250
40  PRINT ''A - SOU BASTANTE GORDO''
50  PRINT ''B - SOU UM POUCO GORDO''
60  PRINT ''C - TENHO ALGUM PESO A MAIS''
70  PRINT ''D - TENHO O PESO CERTO''
80  PRINT ''E - SOU MAIS MAGRO DO QUE DEVIA''
90  GO SUB 1170
100 LET WEIGHT=5*(CODE (A$)-68):IF A$=''E'' THEN LET WEIGHT=0
```

```
110 PRINT WEIGHT
120 GO SUB 1250
130 INK 5: PRINT ''FAÇO EXERCÍCIO, ELEVANDO O RITMO''
140 PRINT ''CARDÍACO PARA 120 OU MAIS,''
150 PRINT ''PELO MENOS O SEGUINTE NÚMERO DE''
160 PRINT ''HORAS POR SEMANA:''
170 INK 7: PRINT
180 PRINT ''A - MENOS DE UM QUARTO DE HORA''
190 PRINT ''B - ENTRE UM QUARTO E TRÊS QUARTOS DE HORA''
200 PRINT ''C - ENTRE TRÊS QUARTOS DE HORA E HORA E MEIA''
210 PRINT ''D - ENTRE UMA HORA E MEIA E DUAS HORAS E MEIA''
220 PRINT ''E - MAIS DE DUAS HORAS E MEIA''
230 GO SUB 1170
240 LET EXERCISE=5*(CODE A$-63)-5: IF A$=''A'' THEN LET
    EXERCISE=0
250 PRINT EXERCISE
260 GO SUB 1250
270 PRINT INK 5;''QUANDO GUIO:'':PRINT
280 PRINT ''A - QUASE NUNCA USO O CINTO DE SEGURANÇA''
290 PRINT ''B - USO O CINTO CERCA DE UM QUARTO DAS VEZES''
300 PRINT ''C - USO O CINTO EM UMA DE CADA DUAS VEZES''
310 PRINT ''D - USO O CINTO QUASE SEMPRE, MAS NEM SEMPRE''
320 PRINT ''E - USO SEMPRE O CINTO''
330 GO SUB 1170
340 LET SEATBELT=2*(CODE A$-65)
350 PRINT SEATBELT
360 GO SUB 1250
370 PRINT INK 5;''CONHEÇO OS PROBLEMAS DE NUTRIÇÃO E TENTO
    COMER DE FORMA SALUTAR:''
380 PRINT
390 PRINT ''A - SEMPRE''
400 PRINT ''B - QUASE SEMPRE''
410 PRINT ''C - BASTANTES VEZES''
420 PRINT ''D - DE VEZ EM QUANDO''
430 PRINT''E - QUASE NUNCA''
440 GO SUB 1170
450 LET DIET=-CODE A$+69
460 PRINT DIET
470 GO SUB 1250
480 PRINT INK 5; ''FUMO (UM CHARUTO CONTA COMO UM CIGARRO)''
490 PRINT
500 PRINT ''A - NADA''
510 PRINT ''B - MENOS DE 15 CIGARROS POR DIA''
520 PRINT ''C - 15 A 25 CIGARROS POR DIA''
530 PRINT ''D - 26 A 42 CIGARROS POR DIA''
```

```
540 PRINT ''E - MAIS DE 42 CIGARROS POR DIA''
550 GO SUB 1170
560 LET SMOKING=-7*(CODE A$-65)
570 PRINT SMOKING
580 GO SUB 1250
590 PRINT INK 5;''ALCOOL - QUANTAS BEBIDAS (EM MÉDIA) POR
    DIA?''
600 PRINT
610 PRINT ''A - NENHUMA''
620 PRINT ''B - MENOS DE 3''
630 PRINT ''C - 3 A 6''
640 PRINT ''D - 7 A 9''
650 PRINT ''E - MAIS DE 9''
660 GO SUB 1170
670 LET DRINK=-30
680 IF A$= ''A'' THEN LET DRINK=0
690 IF A$= ''B'' THEN LET DRINK=1
700 IF A$= ''C'' THEN LET DRINK=DRINK/5
710 IF A$= ''D'' THEN LET DRINK=DRINK/2
720 PRINT DRINK
730 GO SUB 1250
740 PRINT INK 5; ''EM GERAL, QUE STRESS SENTIU NOS ÚLTIMOS
    SEIS MESES''
760 PRINT
770 PRINT ''A - UM STRESS EXTREMO''
780 PRINT ''B - UM STRESS RELATIVO''
790 PRINT ''C - ALGUM STRESS''
800 PRINT ''D - NEUTRO''
810 PRINT ''E - SEM STRESS''
820 GO SUB 1170
830 LET STRESS=INT (2.5*(CODE A$-69))
840 PRINT STRESS
850 GO SUB 1300: CLS
860 PRINT ''OPINIÃO PESSOAL DE MEDICI:''
870 PRINT : INK 6
880 PRINT TAB 8;''PESO:'',WEIGHT
890 PRINT TAB 6;''EXERCÍCIO:'',EXERCISE
900 PRINT TAB 4;''SEGURANÇA DA CONDUÇÃO:'';SEATBELT
910 PRINT TAB 5;''NUTRIÇÃO:'',DIET
920 PRINT TAB 7;''FUMO:'',SMOKING
930 PRINT TAB 7;''ÁLCOOL:'',DRINK
940 PRINT TAB 8;''STRESS:'',STRESS
950 GO SUB 1300
960 LET ANT=WEIGHT+EXERCISE+SEATBELT+DIET+SMOKING+DRINK+
    +STRESS
```

```
970 GO SUB 1300: PRINT
980 PRINT INK 5;''A SUA CLASSIFICAÇÃO GENÉRICA É'',ANT: PRINT
990 INK 7: PRINT ''NUMA ESCALA EM QUE O É UM'',''VALOR
    MÉDIO,'';
1000 PRINT ''A MENOR CLASSIFICAÇÃO É INFERIOR A -80, E''
1010 PRINT''A MAIOR É'',''SUPERIOR A 30''
1020 GO SUB 1300: PRINT
1030 IF ANT<6 AND ANT>-6 THEN LET A$=''MÉDIA'': LET L$=''62 A
     73   72 A 78''
1040 IF ANT<-5 AND ANT>-21 THEN LET A$=''INFERIOR À '': LET
     L$=''60 A 66   65 A 71''
1050 IF ANT<-20 THEN LET A$=''FRACO'': LET L$=''60 OU MENOS
     65 OU MENOS''
1060 IF ANT<-45 THEN LET A$='''MUITO FRACO''
1070 IF ANT<-60 THEN LET A$=''MUITO, MUITO FRACO''
1080 IF ANT>5 AND ANT< 15 THEN LET A$=''BOM'': LET L$=''74 A
     80   79 A 85''
1090 IF ANT>14 THEN LET A$=''MUITO BOM'': LET L$=''81 OU MAIS
     86 OU MAIS''
1100 PRINT ''ISTO INDICA QUE O SEU ESTADO DE SAÚDE É:'';A$
1110 FOR J=1 TO 500: NEXT J: CLS : PRINT : PRINT
1120 PRINT TAB 8;''EXPECTATIVA DE VIDA:''
1130 PRINT TAB 8; INK 5;''HOMEM   MULHER''
1140 PRINT TAB 6; INK 6;L$
1150 STOP
1160 REM ***********************
1170 REM ACEITAR ENTRADA
1180 IF INKEY$<>'' ''THEN GO TO 1180
1190 LET A$=INKEY$
1200 IF A$<''A'' OR A$>''E'' THEN GO TO 1190
1210 PRINT : PRINT INK 6; ''OK, '';A$'' '';
1220 BEEP .4,5*(CODE A$-65): RETURN
1230 REM ***********************
1240 REM ATRASO/ESPAÇAR
1250 FOR J=1 TO 200: NEXT J
1260 CLS
1270 PRINT
1280 PRINT INK 6;''QUAL DAS SEGUINTES OPÇÕES ESTÁ MAIS
     PRÓXIMA DA VERDADE:''
1290 PRINT
1300 FOR J=1 TO 100: NEXT J
1310 BEEP .15,15: BEEP .1,20: RETURN
```

RITA

```
10  REM RITA
20  GO SUB 1380: REM INICIALIZAR
30  GO SUB 1160: REM OPÇÕES DE SAÍDA
40  GO SUB 1250: REM OPÇÕES DE DISCRIMINAÇÃO
50  GO SUB 140: REM PERGUNTAS
60  GO SUB 480: REM TOMAR DECISÃO E ACTUALIZAR BASE DE
    CONHECIMENTOS
70  PRINT INK 6; TAB3;''<ENTER> PARA CONTINUAR'';
80  IF X$<>'' '' THEN INPUT LINE I$: GO TO 50
90  PRINT INK 6;TAB 6;''APRENDIZAGEM''
100 PRINT INK 6;TAB 3;''OU OUTRA TECLA PARA USAR RITA'';
110 INPUT LINE X$:GO TO 50
120 STOP
130 REM ***********************
140 REM PERGUNTAS AO UTILIZADOR
150 CLS
160 PRINT ''EIS OS TEMAS SOBRE OS QUAIS POSSO DISCRIMINAR:'
170 PRINT
180 FOR J=1 TO TT
190 PRINT INK 5;''>   ''; INK 7;A$(J)
200 NEXT J
210 GO SUB 1500
220 IF X$='' '' THEN PRINT INK 6;'' PENSE NUM, E CARREGUE EM
    <ENTER>''
230 IF X$<>'' '' THEN PRINT INK 6;''ESTOU PRONTA A
    DETERMINAR QUAL É''
240 IF X$='' '' THEN INPUT LINE J$
250 LET ADD=.5
260 FOR J=1 TO DQ
270 LET ADD=ADD+ADD
280 GO SUB 1500
290 IF X$<>'' ''AND TT>2 THEN GO TO 390: REM VERIFICAR SE A
    PERGUNTA PODE SER OMITIDA
300 PRINT ''ESCREVA UM NÚMERO ENTRE''
310 PRINT''1 (VERDADEIRO) E 0 (FALSO) ($ PARA TERMINAR)''
320 PRINT : PRINT INK 5;E$(J)
330 INPUT LINE H$:IF H$=''$'' THEN PRINT : PRINT
    ''OBRIGADO'': PRINT : STOP
340 LET C(J)=VAL (H$)
350 LET C(J)=ADD*C(J)
360 NEXT J
370 RETURN
380 REM ***********************
```

```
390 REM VERIFICAR SE PODE OMITIR PERGUNTA
400 LET JUMP=1
410 FOR W=1 TO TT
420 IF ABS (B(W,J)-B(1,J))>.7 THEN LET JUMP=0
430 NEXT W
440 IF JUMP=0 THEN GO TO 300
450 LET C(JUMP)=B(W,J)
460 GO TO 360
470 REM ***********************
480 REM TOMAR DECISÃO
490 FOR J=1 TO TT
500 LET D(J)=0: LET E(J)=0: LET F(J)=0
510 NEXT J
520 LET ADD=.5
530 FOR J=1 TO TT
540 LET ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM JOGAR COM OS VALORES DAS TRÊS LINHAS SEGUINTES PARA
    OBTER OS RESULTADOS MAIS EFICIENTES
570 IF C(X)=B(J,X) THEN LET D(J)=D(J)+1
580 IF ABS (C(X)-B(J,X))<.6*ADD THEN LET E(J)=E(J)+.4
590 IF ABS (C(X)-B(J,X))<1.2*ADD THEN LET F(J)=F(J)+.1
600 NEXT X
610 NEXT J
620 LET A1=1: LET A2=1: LET A3=1
630 LET F1=1: LET F2=1: LET F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN LET F1=D(J): LET A1=J
660 IF E(J)>F2 THEN LET F2=E(J): LET A2=J
670 IF F(J)>F3 THEN LET F3=F(J): LET A3=J
680 NEXT J
690 REM ** ANUNCIAR RESULTADO **
700 PRINT
710 LET CFLG=0
720 PRINT ''O RESULTADO MAIS PROVÁVEL É'''TAB 1; INK 5;A$(A1)
730 IF A2<>A1 THEN PRINT ''O MAIS PROVÁVEL SEGUINTE É''' TAB
    1; INK 5; ;A$(A2): LET CFLG=1
740 IF A3 <>A2 AND A3<>A1 THEN PRINT '''O MAIS PROVÁVEL A
    SEGUIR É''TAB 1; INK 5; A$(A3): LET CFLG=2
750 PRINT
760 PRINT INK 6;''O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO
    (S OU N)?''
770 INPUT LINE F$
780 IF F$<>''S'' AND F$<>''N'' THEN GO TO 770
790 IF F$<>''S'' AND X$<>'' ''THEN RETURN
```

```
800 IF F$=''S'' THEN GO TO 980
810 IF TT=2 AND A1=1 THEN LET A1=2: GO TO 980
820 IF TT=2 THEN LET A1=1: GO TO 980
830 IF CFLG=0 THEN GO TO 890
840 PRINT ''A SEGUNDA OPÇÃO ESTÁ CORRECTA (S OU N)?''
850 INPUT LINE F$
860 IF F$=''N'' THEN GO TO 890
870 IF CFLG=1 THEN A1=A2: GO TO 980
880 IF CFLG=2 THEN A1=A3: GO TO 980
890 GO SUB 1500
900 FOR J=1 TO TT
910 PRINT INK 5;J;''-  ''; INK 7;A$(J)
920 NEXT J
930 PRINT
940 PRINT ''QUAL É A CORRECTA?''
950 INPUT A1
960 IF A1<1 OR A1>TT THEN GO TO 950
970 REM ** EDUCANDO RITA **
972 REM (ACTUALIZAR BASE DE CONHECIMENTOS)
980 FOR J=1 TO DQ
990 IF B(A1,J)<>0 THEN LET B(A1,J)=(C(J)+5*B (A1,J))/6
1000 IF B(A1,J)=0 THEN LET B(A1,J)=C(J)
1010 LET B(A1,J)=INT (.5+(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$='' '' THEN RETURN
1050 FOR J=1 TO TT
1060 PRINT: GO SUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K); INK 5; B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 RETURN
1150 REM ***********************
1160 REM OPÇÕES DE SAÍDA
1170 TT=0
1180 LET TT=TT+1
1190 GO SUB 1500
1200 PRINT INK 6;''ESCREVA NÚMERO DA OPÇÃO'';TT'''
     (<ENTER>PARA TERMINAR)''
1210 INPUT LINE A$(TT)
1215 PRINT : PRINT A$(TT)
```

```
1220 IF A$(TT)='' '' OR TT=51 THEN LET TT=TT-1: RETURN
1230 GO TO 1180
1240 REM ***********************
1250 REM PERGUNTAS DE DISCRIMINAÇÃO
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 LET DQ=0
1310 LET DQ=DQ+1
1320 GO SUB 1500
1330 PRINT INK 6;''ESCREVA NÚMERO DA PERGUNTA
     '';DQ' ''(<ENTER>PARA TERMINAR)''
1340 INPUT LINE E$(DQ)
1345 PRINT : PRINT E$(DQ)
1350 IF E$(DQ,1)='' '' OR DQ=51 THEN LET DQ=DQ-1: RETURN
1360 GO TO 1310
1370 REM ***********************
1380 REM INICIALIZAÇÃO
1382 POKE 23658,8
1384 POKE 23609,40
1386 POKE 23692,255
1390 BORDER 1: PAPER 1: INK 7: CLS
1400 REM REDUZIR ARRAYS NA LINHA SEGUINTE DE ACORDO COM AS
     NECESSIDADES
1410 DIM A$(50,15): DIM B(50,50): DIM C(50): DIM D(50): DIM
     E$(50,40): DIM E(50): DIM F(50)
1420 LET X$='' ''
1430 PRINT ''CARREGUE NUMA TECLA E EM <ENTER> SE QUISER
     OBSERVAR A BASE DE CONHECIMENTOS ACTUALIZADA DEPOIS DE
     CADA EXECUÇÃO;''
1470 PRINT ''CARREGUE SÓ EM <ENTER>SE NÃO QUISER''
1470 INPUT LINE U$
1480 CLS
1490 RETURN
1500 PRINT INK 2:''-------------------------------
---------------''
1502 POKE 23692,255
1510 RETURN
```

HASTE

```
5   PAPER 1: INK 7: BORDER 1
10  REM HASTE
```

```
15  POKE 23609,40: POKE 23658,8
20  DIM A$(255,20): DIM B$(255,50)
30  LET F$='' '': LET K=0: LET F=0: CLS
40  REM ***********************
50  LET FLAG=0
60  INPUT ''>  ''; LINE D$: PRINT D$: LET K=K+1
70  IF D$='' '' THEN STOP
80  IF D$(TO 1)=''?'' THEN GO TO 200
90  LET E=0
100 LET E=E+1
110 IF D$(E TO E)=''*'' THEN GO TO 140
120 IF E<LEN D$ THEN GO TO 100
130 PRINT ''INVALID ENTRY'': BEEP .5,0: GO TO 50
140 IF FLAG=3 THEN RETURN
150 LET F=F+1: IF F=256 THEN STOP
160 LET A$(F)=D$(TO E-1)+''@''
170 LET B$(F)=D$(E+1 TO)+''@''
175 BEEP .1,10: BEEP .3,15
180 GO TO 50
190 REM ***********************
200 REM INTERROGAR
210 LET FLAG=4: LET TRUE=0
220 IF D$(LEN D$ TO)=''/'' THEN LET FLAG=3
230 FOR J=1 TO LEN(D$)-5
240 IF D$(J TO J+4)='' AND '' THEN LET FLAG=15: LET TRUE=J
250 NEXT J
260 IF FLAG=5 THEN GO TO 410
270 IF D$(TO 3)=''?/*/'' THEN LET FLAG=1
280 IF D$(TO 4)=''?/*/'' THEN LET FLAG=2
290 IF FLAG=3 THEN GO SUB 90: LET F$=D$(2 TO E-1)
300 IF FLAG=1 THEN LET F$=D$(4 TO)
310 LET E=0
320 LET E=E+1
330 IF CODE A$(E)=32 AND FLAG=4 AND TRUE=0 THEN PRINT
    ''FALSE'': BEEP .1,10: BEEP .3,15
340 IF CODE A$(E)=32 THEN GO TO 520
345 LET P$=A$(E): LET Q$=B$(E): GO SUB 600: GO SUB 650
350 IF FLAG=4 AND ''?''+P$(E)+''*''+Q$=D$ THEN PRINT «TRUE»:
    BEEP .1,10: BEEP .3,15: LET TRUE=1
360 IF FLAG=3 AND F$=P$ THEN PRINT Q$
370 IF FLAG=2 THEN PRINT P$;''*'';Q$
380 IF FLAG=1 AND F$=Q$ THEN PRINT P$
385 BEEP .05,0
390 IF E<K THEN GO TO 320
400 REM ***********************
```

```
410 LET F$=D$(4 TO TRUE-1): LET G$=D$(TRUE+7 TO)
420 LET E=0
430 LET E=E+1
440 IF CODE A$(E)=32 THEN GO TO 520
450 LET Q$=B$(E): GO SUB 650 : IF Q$=F$ THEN GO TO 470
460 IF E<K THEN GO TO 430
470 LET H=0
480 LET H=H+1
490 IF CODE B$(H)=32 THEN GO TO 460
500 LET Q$=B$(H): GO SUB 650: LET P$=A$(E): GO SUB 600: LET
    W$=P$: LET P$=A$(H): GO SUB 600: IF Q$=G$ AND W$=P$ THEN
    PRINT W$
510 IF E<K THEN GO TO 480
520 PRINT INK 6;TAB 5;''>END OF ANSWER<''
530 GO TO 50
600 LET T=0
610 LET T=T+1
620 IF P$(T TO T)=''@'' THEN LET P$=P$(TO T-1): RETURN
650 LET T=0
660 LET T=T+1
670 IF Q$(T TO T)=''@'' THEN LET Q$=Q$ (TO T-1): RETURN
675 BEEP .02,0
680 GO TO 660
```

EASLE

```
10  REM EASLE
20  BORDER 1: PAPER 1: INK 7: CLS
25  POKE 23609,60
27  POKE 23658,8
30  DIM X(40): DIM Y(40): DIM Z(40)
40  PRINT
50  INPUT ''>   ''LINE A$
55  PRINT ''>  ''A$: BEEP .4,10
60  IF A$='' '' THEN BEEP 1,-15: STOP
70  REM ***********************
80  FOR J=1 TO 40
90  LET X(J)=0: LET Y(J)=0: LET Z(J)=0
100 NEXT J
110 REM ***********************
120 LET R=0: LET S=0: LET T=0: LET CFIRST=0: LET CSECND=0: LET
    EDGE=0
130 FOR J=1 TO LEN A$
140 LET B$=A$(J)
```

```
150 IF B$=''('' THEN LET S=S+1: LET Z(S)=J: IF T=0 THEN LET
    CFIRST=J
160 IF B$='')'' THEN LET T=T+1: LET Y(T)=J: IF CSECND<>0 AND
    EDGE=0 THEN LET EDGE=J-1
170 IF T=1 AND B$='')'' THEN LET CSECND=J
180 IF B$= '' '' THEN LET R=R+1: LET X(R)=J
190 NEXT J
200 IF S=T THEN GO TO 260: REM EQUILÍBRIO ( )
210 BEEP .4,-10: IF S<T THEN PRINT '' ->MISSING (''
220 IF S>T THEN PRINT '' ->MISSING )''
230 INPUT ''>   ''; LINE B$
235 PRINT ''>   '';B$
240 LET A$=A$+B$
250 GO TO 80
260 LET FLAG=0
270 IF A$(TO 5)=''CAR ('' THEN LET FLAG=1: GO SUB 420: GO TO
    340
280 IF A$(TO 5)=''CDR ('' THEN LET FLAG=2: GO SUB 470: GO TO
    340
290 IF A$(TO 6)=''CONS ('' THEN LET FLAG=3: GO SUB 550: GO TO 340
300 IF A$(TO 6)=''ATOM ('' THEN LET FLAG=4: GO SUB 690: GO TO 340
310 IF A$(TO 4)=''EQ ('' THEN LET FLAG=5: GO SUB 780: GO TO
    340
320 IF A$(TO 6)=''NULL ('' THEN FLAG=6: GO SUB 920
340 IF FLAG<>0 THEN GO SUB 360
350 GO TO 40
360 REM ** RESPOSTA **
370 PRINT ''   VALUE IS...''
380 IF B$<>''( )'' THEN PRINT''  '';B$
390 IF B$=''( )'' THEN PRINT''  NIL''
400 BEEP .3,20: RETURN
410 REM ***********************
420 REM ** CAR **
430 IF S=2 THEN LET B$=A$(Z(2)+1 TO X(2))
440 IF S>2 THEN LET B$=A$(CFIRST TO CSECND)
450 RETURN
460 REM ***********************
470 REM ** CDR **
480 GO SUB 420
490 LET LB=LEN B$+7
495 LET KK=LB+EDGE2: IF KK>LEN A$ THEN LET KK=LEN A$
500 LET B$=''(''+A$(LB TO KK)
510 IF B$(LEN B$-1 TO )=''))'' THEN LET B$=B$( TO LEN B$-1)
520 IF B$(2)='' '' THEN LET B$=''(''+B$(3 TO)
530 RETURN
```

```
540 REM ***********************
550 REM ** CONS **
560 LET B$=A$(7 TO)
570 LET J=0
580 IF B$(1)=''('' THEN LET J=1
590 LET J=J+1
600 IF B$(J)=''('' THEN GO TO 630
610 IF J<LEN B$ THEN GO TO 590
620 LET B$=''  >CONS ERROR<'':RETURN
630 LET LB=LEN B$-1
640 LET B$=''(''+B$(TO J-1)+B$(J+1 TO)
650 LET B$=B$ (TO LB)
660 IF B$(LEN B$-1 TO)='' )'' THEN LET B$=B$ (TO LEN B$-
    2)+'')''
670 RETURN
680 REM ***********************
690 REM ** ATOM **
700 A$=A$(7 TO LEN A$-1)
710 LET J=0: LET B$=''NIL''
720 LET J=J+1
730 IF A$(J)='' '' OR A$(J)=''('' THEN RETURN
740 IF J<LEN A$ THEN GO TO 720
750 LET B$=''T''
760 RETURN
770 REM ***********************
780 REM ** EQ **
790 LET A$=A$(5 TO)
800 LET A$=A$ (TO LEN A$-1)
810 LET J=0: LET B$=''NIL''
820 LET J=J+1
830 IF A$(J)='')'' THEN RETURN
840 IF A$(J)='' '' THEN GO TO 870
850 IF J<LEN A$ THEN GO TO 820
860 RETURN
870 LET C$=A$ (TO J-1)
880 LET A$=A$(J+1 TO)
890 IF C$=A$ THEN LET B$=''T''
900 RETURN
910 REM ***********************
920 REM ** NULL **
930 LET B$=''NIL''
940 IF A$=''NULL ( )'' THEN BEEP. 4,-20: PRINT ''ILLEGAL -
    NULL NEEDS ARGUMENT'': LET B$='' ''
950 IF A$=''NULL (( ))'' THEN LET B$=''T''
960 RETURN
```

PROLOG-A

```
5   PAPER 1: BORDER 1: INK 7
10  REM PROLOG-A (SIMPLE FRONT-END)
15  POKE 23609,40: POKE 23658,8
20  REM ENTRADAS EM MAIÚSCULAS
30  GO TO 50
40  PRINT ''NO (MORE) ANSWERS'': RETURN
50  GO SUB 3270: REM INICIALIZAR
60  REM ********************
70  PRINT
75  PRINT AT 0,0;''
                  ''
80  INPUT ''&.''; LINE J$
90  IF J$='' '' THEN STOP
92  CLS
95  PRINT INK 7; PAPER 0; FLASH 1; AT 0,10;''PLEASE WAIT''
100 IF J$=''LIST ALL'' THEN GO SUB 1860: GO TO 70
110 IF J$(TO 5)=''LIST '' THEN LET J$=J$(5 TO)+'' '':GO SUB
    990: GO TO 70
120 IF J$(LEN J$ TO)<>'')'' THEN PRINT''1.'': INPUT L$: LET
    J$=J$+L$: GO TO 120
130 LET LJ=LEN J$
140 LET J$=J$(TO LJ-1)+'' '':REM TIRAR ) FINAL, SUBSTITUIR POR
    ESPAÇO
150 LET LJ=LEN(J$)
160 LET FLAG=0
170 IF J$(TO 4)=''ADD(''THEN LET J$=J$(5 TO): LET FLAG=1
180 LET RULEFLAG=0: LET PLUSFLAG=0: LET ARITHFLAG=0
190 FOR R=1 TO LEN J$
195 IF (R+3)>LEN J$ THEN GO TO 205
200 IF J$(R TO R+3)='' IF '' THEN RULEFLAG=R: LET FLAG=6
205 IF (R+4)>LEN J$ THEN GO TO 215
210 IF J$(R TO R+4)='' AND '' THEN LET PLUSFLAG=R
215 IF (R+3)>LEN J$ THEN GO TO 225
220 IF J$(R TO R+3))=''SUM('' THEN LET ARITHFLAG=1
225 IF (R+5)>LEN J$ THEN GO TO 235
230 IF J$(R TO R+5)=''TIMES('' THEN LET ARITHFLAG=2
235 IF (R+5)>LEN J$ THEN GO TO 245
240 IF J$(R TO R+5)=''LESS '' THEN LET ARITHFLAG=3
245 IF (R+2)>LEN J$ THEN GO TO 260
250 IF J$(R TO R+2)='' INT '' THEN LET ARITHFLAG=4
260 NEXT R
265 IF LEN J$<3 THEN GO TO 275
270 IF J$(TO 3)=''IS('' THEN LET J$=J$(4 TO): LET FLAG=2
```

```
275 IF LEN J$<10 THEN GO TO 285
280 IF J$(TO 10)=''WHICH(X : '' THEN LET J$=J$(11 TO): LET
    FLAG=3
285 IF LEN J$<16 THEN GO TO 300
290 IF J$(TO 16)=''WHICH((X Y) : X '' THEN LET J$=J$( 17 TO):
    LET FLAG=4
300 IF FLAG=0 THEN PRINT''SYNTAX ERROR'':GO TO 70
310 LET LJ=LEN J$
320 REM ENVIAR PARA SUBROTINAS RELEVANTES
330 IF PLUSFLAG<>0 THEN GO SUB 1950: GO TO 70:REM CODIFICAR
    REGRA CONTENDO AND
340 IF RULEFLAG<>0 AND FLAG<>5 THEN GO SUB 1110: REM
    CODIFICAR REGRA
350 IF ARITHFLAG<>0 THEN GO SUB 2430: GO TO 70: REM
    ARITMÉTICA
360 IF J$(LEN J$-2 TO)='' X '' OR J$(LEN, J$-2)= '' Y '' THEN
    LET J$=J$( TO LJ-2)+'' ''
370 LET LJ=LEN J$
380 IF FLAG=1 THEN GO SUB 440: REM ADD
390 IF FLAG=2 THEN GO SUB 520: REM IS
400 IF FLAG=3 THEN GO SUB 610: REM WHICH
410 IF FLAG=4 THEN GO SUB 830: REM WHICH2
420 GO TO 70
430 REM ***********************
440 REM    ADD
450 LET K=0
460 LET K=K+1
470 IF CODE Z$(K)=32 THEN LET Z$(K)=J$+''@'': BEEP .1,10:
    BEEP .3,15: RETURN
480 IF K<500 GO TO THEN 460
490 PRINT INK 5; FLASH 1;''MEMORY FULL'': BEEP .5,15
500 RETURN
510 REM ***********************
520 REM    IS
530 LET K=0
540 LET K=K+1
550 IF CODE Z$(K)=32 THEN GO TO 580
560 LET A$=Z$(K): GO SUB 8500: LET J$=J$( TO LEN J$=-1): IF
    A$=J$+'' ''THEN PRINT INK 6;''YES'': GO TO 590
570 IF K<500 GO TO 540
580 PRINT INK 6;''NO''
590 BEEP .1,10: BEEP .3,15: RETURN
600 REM ***********************
610 REM        WHICH
620 IF J$(1)=''X'' THEN GO TO 710
```

```
630 LET J$=J$( TO LJ-1)
640 LET K=0
650 LET K=K+1
660 IF CODE Z$(K)=32 THEN GO TO 690
670 LET A$=Z$(K): GO SUB 8500: IF J$=A$ (TO LEN J$) THEN PRINT
    INK 6;A$(LEN J$ TO)
680 IF K<500 THEN GO TO 650
690 GO SUB 40
695 BEEP .1,10: BEEP .3,15
700 RETURN
710 REM * PERGUNTA COMEÇA POR X*
720 LET J$=J$(3 TO LEN J$)
730 LET LJ=LEN(J$)
740 LET K=0
750 LET K=K+1
760 IF CODE Z$(K)=32 THEN GO TO 800
770 LET A$=Z$(K): GO SUB 8500: LET Q$=A$(LEN A$-LJ+1 TO)
780 IF Q$=J$ THEN PRINT INK 6; A$(TO LEN A$-LJ-1)
790 IF K<500 THEN GO TO 750
800 GO SUB 40
805 BEEP .1,10: BEEP .3,15
810 RETURN
820 REM ***********************
830 REM   WHICH2
840 LET J$=J$(TO LJ-2)
850 LET LJ=LEN J$
860 LET K=0
870 LET K=K+1
880 IF CODE Z$(K)=32 THEN GO TO 960
890 LET LFLAG=0
895 LET A$=Z$(K)
900 FOR L=1 TO 30-LJ)
910 IF A$(L TO L+LJ-1)=J$ THEN LET LFLAG=L
920 NEXT L
930 IF LFLAG=0 THEN GO TO 950
935 GO SUB 8500
940 PRINT A$(TO LFLAG-2);A$(LFLAG+LJ TO)
945 BEEP .05,0
950 IF K<500 THEN GO TO 870
960 GO SUB 40
965 BEEP .1,10: BEEP .3,15
970 RETURN
980 REM ***********************
990 REM   LISTAR
1000 LET K=0
```

```
1010 LET K=K+1
1020 IF CODE Z$(K)=32 THEN RETURN
1030 LET LFLAG=0
1040 FOR L=1 TO LEN Z$(K)-LEN J$
1050 LET A$=Z$(K): IF A$(L TO L+LEN J$-1)= J$ THEN LFLAG=1
1060 NEXT L
1070 IF LFLAG=1 THEN LET A$=Z$(K): GO SUB 8500: PRINT INK 6;A$
1075 BEEP .05,0
1080 IF K<500 GO TO 1010
1090 RETURN
1100 REM **********************
1110 REM   FORMAR REGRAS
1120 LET R=RULEFLAG
1130 LET E$=J$(TO R): LET F$=J$(R+4 TO)
1140 IF E$(1)<>''X'' THEN PRINT INK 5; FLASH 1;''RULE
     ERROR'': GO TO 70
1150 REM A LINHA SEGUINTE DETECTA ENTRADAS COMO X COME Y SE X
     É UM Y
1160 IF F$(LEN F$-1 TO)=''Y'' THEN GO TO 1390
1170 PRINT INK 7;TAB 8 ;''COMPILING RULE''
1180 FOR T=1 TO 100
1190 LET R$(T)='' ''
1200 NEXT T
1210 LET E$=E$(3 TO): LET F$=F$(3 TO LEN F$)
1220 LET K=0: LET RR=0
1230 LET K=K+1
1240 IF CODE Z$ (K)=32 THEN GO TO 1300
1245 LET A$=Z$(K): GO SUB 8500
1250 IF A$(LEN A$-LEN F$+1 TO)<>F$ THEN GO TO 1370
1260 LET RR=RR+1
1270 LET R$(RR)=A$(TO LEN A$-LEN F$)+E$+''@''
1280 PRINT ''>  '';: LET A$=R$(RR): GO SUB 8500: PRINT INK
     6;A$
1285 BEEP .1,10: BEEP.3,15
1290 GO TO 1230
1300 IF RR=0 THEN RETURN
1310 LET RC=0
1320 LET RC=RC+1
1330 LET Z$(K)=R$(RC)
1340 IF K<500 THEN LET K=K+1
1350 IF RC<RR THEN GO TO 1320
1360 RETURN
1370 IF K<500 THEN GO TO 1230
1380 RETURN
1390 REM ** REGRA COM DUAS VARIÁVEIS *
```

```
1400 FOR T=1 TO 100
1410 LET R$(T)='' ''
1420 NEXT T
1430 LET K=0: LET RR=0
1440 IF K=500 THEN RETURN
1450 LET K=K+1
1460 IF CODE Z$(K)=32 THEN GO TO 1770
1470 REM DIVIDIR EM 3 PALAVRAS
1480 LET Q$=Z$(K)
1490 LET J=0
1500 LET J=J+1
1510 IF Q$(J TO J)='' '' THEN GO TO 1540
1520 IF J<LEN Q$ THEN GO TO 1500
1530 PRINT INK 5;''RULE COMPILING ERROR'': GO TO 70
1540 LET A$=Q$(TO J)
1550 LET Q$=Q$(J+1 TO)
1560 LET J=0
1570 LET J=J+1
1580 IF Q$(J)='' ''THEN GO TO 1610
1590 IF J<LEN Q$ THEN GO TO 1570
1600 PRINT INK 5;''RULE COMPILING ERROR'': GO TO 70
1610 LET B$=Q$(TO J)
1620 LET Q$=Q$(J+1 TO)
1630 LET J=0
1640 LET J=J+1
1650 IF Q$(J)= '' ''THEN GO TO 1680
1660 IF J<LEN Q$ THEN GO TO 1640
1670 PRINT INK 5;''RULE COMPILING ERROR'':GO TO 70
1680 PRINT INK 7;TAB 8;''COMPILING RULE''
1690 LET C$=Q$(TO J)
1700 LET M$=F$(3 TO 3+LEN B$-1)
1710 IF B$<>M$ THEN GO TO 1440
1720 LET RR=RR+1
1730 LET N$=E$(3 TO LEN E$-2)
1740 LET R$(RR)=A$+N$+C$''@''
1750 PRINT''>   '';: LET A$=R$(RR): GO SUB 8500: PRINT INK 6;A$
1755 BEEP .1,10: BEEP .3,15
1760 GO TO 1440
1770 IF RR=0 THEN RETURN
1780 LET M=0
1790 LET M=M+1
1800 IF M>RR THEN RETURN
1810 LET Z$(K)=R$(M)
1820 IF K=500 THEN PRINT INK 5; FLASH 1;''OUT OF MEMORY'':
     BEEP .5,15: GO TO 70
```

```
1830 GO TO 1790
1850 REM **********************
1860 REM LISTAR TUDO
1870 PRINT
1880 LET K=0
1890 LET K=K+1
1900 IF CODE Z$(K)=32 THEN RETURN
1910 LET A$=Z$(K): GO SUB 8500: PRINT INK 6;A$
1915 BEEP .05,0
1920 IF K<500 THEN GO TO 1890
1930 RETURN
1940 REM **********************
1950 REM FORMAR REGRAS COM ''AND'' DO SEGUINTE TIPO:
1960 REM (X COME Y SE X É UMA AVE E Y VEM EM CAIXAS)
1970 REM A DECLARAÇÃO X DEVE OCORRER NA LISTA PRECEDENDO Y EM
     TODOS OS CASOS
1980 REM DIVIDIR EM SECÇÕES
1990 LET J$=J$(2 TO): REM SEPARAR ''X''
2000 PRINT INK 7; TAB 8: ''COMPILING RULE''
2010 LET J=1
2020 LET J=J+1
2030 IF J$(J)='' '' THEN GO TO 2060
2040 IF J<LEN J$ THEN GO TO 2020
2050 PRINT INK 5;''RULE COMPILING ERROR'': RETURN
2060 LET I$=J$(TO J): REM RELAÇÃO 1
2070 LET J$=J$(J+7 TO): REM INÍCIO DA SEGUNDA RELAÇÃO
2080 LET J=1: COUNT=0
2090 LET J=J+1
2100 IF J$(J)='' '' THEN LET COUNT=COUNT+1
2110 IF COUNT=2 THEN GO TO 2140
2120 IF J<LEN J$ THEN GO TO 2090
2130 PRINT INK 5;''RULE COMPILING ERROR'':RETURN
2140 LET B$=J$(TO J): REM DECLARAÇÃO 1
2150 LET C$=J$(J+6 TO): REM DECLARAÇÃO 2
2160 IF C$='' ''THEN PRINT ''RULE COMPILING ERROR'': RETURN
2170 REM PERCORRER BASE DE DADOS
2180 FOR T=1 TO 200
2190 LET R$(T)='' ''
2200 NEXT T
2210 LET R1=0: LET R2=99
2220 LET K=0
2230 LET K=K+1
2240 IF CODE Z$(K)=32 THEN GO TO 2310
2250 IF R1=99 OR R2=200 THEN PRINT''MEMORY SHORTAGE'':GO TO
     2310
```

```
2260 LET LB=LEN B$
2270 LET A$=Z$(K): GO SUB 8500: IF A$-LB+1 TO)=B$ THEN LET
     R1=R1+1: LET R$(R1)=A$(TO LEN A$-LB)+''@''
2280 LET LC=LEN C$
2290 LET A$=Z$(K): GO SUB 8500: IF A$(LEN A$-LC+1 TO)=C$ THEN
     LET R2=R2+1: LET R$(R2)=A$( TO LEN A$-LC)+''@''
2300 IF K<500 THEN GO TO 2230
2310 IF CODE R$(100)=32 THEN PRINT INK 5;''STATEMENT 2 OF
     INPUT NOT IN DATABASE'': BEEP .1,10: BEEP .1,15:
     RETURN
2320 REM CODIFICAR REGRAS
2330 LET R1=0: LET R2=99
2340 LET R2=R2+1
2350 LET R1=R1+1
2360 IF R$(R1)='' ''OR R$(R2)='' '' THEN GO TO 2370
2362 GO SUB 8600: GO SUB 8700
2365 LET Z$(K)=W$+I$+V$+'' ''+''@''
2370 PRINT ''>'';: LET A$=Z$(K): GO SUB 8500: PRINT INK 6;A$
2375 BEEP .1,10: BEEP .3,15
2380 LET K=K+1
2390 IF CODE R$(R2+1)<>32 GO TO THEN 2340
2400 IF R$(R1+1)<>32 THEN GO TO 2350
2410 RETURN
2420 REM ********************
2430 REM   ARITMÉTICA
2440 LET LJ=LEN J$
2450 IF ARITHFLAG<3 THEN GO SUB 2490
2460 IF ARITHFLAG=3 THEN GO SUB 2890
2470 IF ARITHFLAG=4 THEN GO SUB 3080
2480 RETURN
2490 REM *** SUM TIMES ***
2500 LET J$=J$(5 TO LJ)
2510 IF J$(TO 2)=''S('' THEN LET J$=J$(3 TO)
2520 LET LJ=LEN J$
2530 LET K=0
2540 LET K=K+1
2550 IF J$(K)='' ''THEN LET A$=J$ (TO K-1): LET J$=J$(K+1 TO):
     GO TO 2580
2560 IF K<LJ THEN GO TO 2540
2570 PRINT INK 5; TAB 6:''ARITHMETIC ERROR'': RETURN
2580 LET LJ=LEN J$
2590 LET K=0
2600 LET K=K+1
2610 IF J$(K)='' '' THEN LET B$=J$ ( TO K-1): LET J$=J$(K+1
     TO): GO TO 2640
```

```
2620 IF K<LJ THEN GO TO 2600
2630 PRINT INK 5; TAB 6;''ARITHMETIC ERROR'': RETURN
2640 LET LJ=LEN J$
2650 LET K=0
2660 LET K=K+1
2670 IF J$(K)='')'' THEN LET C$=J$ (TO K-1): GO TO 2700
2680 IF K<LJ THEN GO TO 2660
2690 PRINT INK 5;TAB 2;''ERROR [TOO MANY VARIABLES]'':
     RETURN
2700 LET AN=0: LET BN=0: LET CN=0
2710 IF CODE A$>57 THEN LET AN=1
2720 IF CODE B$>57 THEN LET BN=2
2730 IF CODE C$>57 THEN LET CN=4
2740 LET GUIDE=AN+BN+CN: IF GUIDE=3 OR GUIDE=5 OR GUIDE=6
     THEN GO TO 2690
2750 IF ARITHFLAG=2 THEN GO TO 2820: REM TIMES
2760 IF GUIDE>0 THEN GO TO 2790
2770 IF VAL A$+VAL B$=VAL C$ THEN PRINT INK 6;''YES'': BEEP
     .1,10: BEEP .3,15: RETURN
2780 PRINT INK 6;''NO'': BEEP .1,10: BEEP .3,15: RETURN
2790 IF GUIDE=1 THEN PRINT VAL C$-VAL B$: GO SUB 40: RETURN
2800 IF GUIDE=2 THEN PRINT VAL C$-VAL A$ : GO SUB 40: RETURN
2810 PRINT INK 6;VAL A$+VAL B$ : GO SUB 40: RETURN
2820 REM ** TIMES **
2830 IF GUIDE>0 THEN GO TO 2860
2840 IF VAL A$*VAL B$ =VAL C$ THEN PRINT INK 6;''YES'': BEEP
     .1,10: BEEP .3,15: RETURN
2850 PRINT INK 6; ''NO'': BEEP .,10: BEEP .3,15: RETURN
2860 IF GUIDE=1 THEN PRINT VAL C$/VAL B$: GO SUB 40: RETURN
2870 IF GUIDE=2 THEN PRINT VAL C$/VAL A$: GO SUB 40: RETURN
2880 PRINT INK 6;VAL A$*VAL B$ : GO SUB 40: RETURN
2890 REM ***********************
2900 LET NF=0
2910 IF CODE J$<58 THEN LET NF=1: REM NUMEROS
2920 LET COUNT=0
2930 LET K=0
2940 LET K=K+1
2950 IF J$(K)='' '' THEN LET COUNT=COUNT+1
2960 IF COUNT=2 THEN GO TO 3000
2970 IF K<LEN J$ THEN GO TO 2940
2980 PRINT INK 5;TAB 7;''COMPARISON ERROR''
2990 RETURN
3000 LET B$=J$(K+1 TO)
3010 LET A$=J$( TO K-6)
3020 IF NF=1 THEN GO TO 3050
```

```
3030 IF A$<B$ THEN PRINT INK 6;''YES'': RETURN
3040 PRINT INK 6;''NO'': BEEP .1,10: BEEP .3,15: RETURN
3050 REM * NÚMEROS *
3060 IF VAL A$<VAL B$ THEN PRINT INK 6;''YES'': BEEP .1,10:
     BEEP .3,15: RETURN
3070 PRINT INK 6;''NO'': BEEP .1,10: BEEP .3,15: RETURN
3080 REM ********* INT **********
3090 IF J$(LEN J$-1 TO)=''X'' THEN GO TO 3190
3100 LET K=0
3110 LET K=K+1
3120 IF J$(K)='' '' THEN GO TO 3160
3130 IF K<LEN J$ THEN GO TO 3110
3140 PRINT INK 5;TAB 7;''ARITHMETIC ERROR''
3150 RETURN
3160 LET A=VAL (J$(TO K-1))
3170 IF INT(A)=A THEN PRINT INK 6; ''YES'': BEEP .1,10: BEEP
     .3,15: RETURN
3180 PRINT INK 6;''NO'': BEEP .1,10: BEEP .3,15: RETURN
3190 LET K=0
3200 LET K=K+1
3210 IF J$(K)='' '' THEN GO TO 3240
3220 IF K<LEN J$ THEN GO TO 3200
3230 PRINT INK 5;TAB 7;''ARITHMETIC ERROR'':RETURN
3240 PRINT INK 6;INT (VAL (J$( TO K-1)))
3250 RETURN
3260 REM ***********************
3270 REM INICIALIZAR
3280 CLS
3290 DIM Z$(500,30): DIM R$(200,30)
3300 RETURN
8500 LET T=1
8505 IF A$(T)=''@'' THEN GO TO 8530
8510 LET D$=A$ ( TO T)
8520 LET T=T+1: GO TO 8505
8530 LET A$=D$: RETURN
8600 LET B$=R$(R17: LET T=1
8605 IF B$(T)=''@'' THEN GO TO 8630
8610 LET W$=B$( TO T)
8620 LET T=T+1: GO TO 8605
8630 RETURN
8700 LET B$=R$(R2): LET T=1
8705 IF B$(T)=''@'' THEN GO TO 8730
8710 LET V$=B$( TO T)
8720 LET T=T+1: GO TO 8705
8730 RETURN
```

SSLISP

```
10  REM S.S.LISP
20  GO TO 3450: REM INICIALIZAÇÃO
30  REM ***********************
40  LET A$=A$(E TO LEN A$-1)
50  RETURN
60  REM ***********************
70  PRINT
80  LET NN=0
90  INPUT LINE A$: PRINT '': '';A$
100 IF A$='' '' THEN BEEP 1,-15: STOP: REM <RETURN> PARA TER-
    MINAR
110 REM ***********************
120 FOR J=1 TO 12
130 LET X(J)=0: LET Y(J)=0: LET Z(J)=0
140 NEXT J
150 REM ***********************
160 LET R=0: LET S=0: LET T=0: LET CFIRST=0: LET CSECND=0: LET
    EDGE=0
170 FOR J=1 TO LEN A$
180 LET B$=A$(J)
190 IF B$=''(''THEN LET S=S+1: LET Z(S)=J: IF T=0 THEN LET
    CFIRST=J
200 IF B$='')'' THEN LET T=T+1: LET Y(T)=J: IF CSECND<>0 AND
    EDGE=0 THEN LET EDGE=J-1
210 IF T=1 AND B$='')'' THEN LET CSECND=J
220 IF B$='' '' THEN LET R=R+1:LET X(R)=J
230 NEXT J
240 IF S=T THEN GO TO 300: REM EQUILÍBRIO ( )
250 IF S<T THEN PRINT '' ->MISSING (''
260 IF S>T THEN PRINT '' ->MISSING )''
270 INPUT ''+ '';LINE B$
280 LET A$=A$+B$
290 GO TO 120
300 IF NWDS=0 OR NN=1 THEN GO TO 370
310 LET M$=A$ (TO X(1)-1)
320 FOR J=1 TO NWDS
330 IF M$=N$(J) THEN LET A$=O$(J)+A$(LEN N$(J))+1 TO)
340 NEXT J
350 LET NN=1
360 GO TO 120
370 LET FLAG=0: LET B$=''NIL''
380 IF A$( TO 4)=''EQ ('' THEN LET FLAG=5: GO SUB 1200: GO TO
    760
```

```
390 IF A$( TO 5)=''CAR ('' THEN LET FLAG=1: GO SUB 840: GO TO
    760
400 IF A$( TO 5)=''CDR ('' THEN FLAG=2: GO SUB 890: GO TO 760
410 IF A$( TO 5)=''MEMQ '' THEN LET FLAG=8: GO SUB 1510: GO TO
    760
420 IF A$( TO 5)=''MAX ('' THEN LET FLAG=19: GO SUB 2600: GO
    TO 760
430 IF A$( TO 5)=''MIN ('' THEN LET FLAG=20: GO SUB 2700: GO
    TO 760
440 IF A$( TO 6)=''CONS ('' THEN LET FLAG=3: GO SUB 970: GO TO
    760
450 IF A$( TO 6)=''ATOM ('' THEN LET FLAG=4: GO SUB 1110: GO
    TO 760
460 IF A$( TO 6)=''NULL ('' THEN LET FLAG=6: GO SUB 1330: GO
    TO 760
470 IF A$( TO 6)=''LIST ('' THEN LET FLAG=12: GO SUB 2310: GO
    TO 760
480 IF A$( TO 6)=''ADD1 ('' THEN LET FLAG=14: GO SUB 2180: GO
    TO 760
490 IF A$( TO 6)=''SUB1 ('' THEN LET FLAG=15: GO SUB 2230: GO
    TO 760
500 IF A$( TO 6)=''EXPT ('' THEN LET FLAG=18: GO SUB 2560: GO
    TO 760
510 IF A$( TO 6)=''PLUS ('' THEN LET FLAG=21: GO SUB 2820: GO
    TO 760
520 IF A$( TO 6)=''ONEP ('' THEN LET FLAG=31: GO SUB 2280: GO
    TO 760
530 IF A$( TO 7)=''ZEROP ('' THEN LET FLAG=16: GO SUB 2280: GO
    TO 760
540 IF A$( TO 7)=''EQUAL ('' THEN LET FLAG=11: GO SUB 2000: GO
    TO 760
550 IF A$( TO 7)=''MINUS ('' THEN LET FLAG=22: GO SUB 2870: GO
    TO 760
560 IF A$( TO 7)=''RECIP ('' THEN LET FLAG=24: GO SUB 2960: GO
    TO 760
570 IF A$( TO 7)=''TIMES ('' THEN LET FLAG=26: GO SUB 3070: GO
    TO 760
580 IF A$( TO 7)=''LESSP ('' THEN LET FLAG=28: GO SUB 3160: GO
    TO 760
590 IF A$( TO 8)=''DEFINE ('' THEN LET FLAG=13: GO SUB 3300:
    GO TO 760
600 IF A$( TO 8)=''APPEND ('' THEN LET FLAG=9: GO SUB 1700: GO
    TO 760
610 IF A$( TO 8)=''MEMBER ('' THEN LET FLAG=7: GO SUB 1380: GO
    TO 760
```

```
620 IF A$( TO 8)=''MINUSP ('' THEN LET FLAG=29: GO SUB 3200:
    GO TO 760
630 IF A$( TO 9)=''REVERSE ('' THEN LET FLAG=10: GO SUB 1760:
    GO TO 760
640 IF A$( TO 9)=''NUMBERP ('' THEN LET FLAG=30: GO SUB 3250:
    GO TO 760
650 IF A$( TO 10)=''QUOTIENT ('' THEN LET FLAG=23: GO SUB
    2920: GO TO 760
660 IF A$( TO 10)=''GREATERP ('' THEN LET FLAG=27: GO SUB
    3120: GO TO 760
670 IF A$( TO 11)=''REMAINDER (''THEN LET FLAG=25: GO SUB
    3030: GO TO 760
680 IF A$( TO 12)=''DIFFERENCE ('' THEN LET FLAG=17: GO SUB
    2350
760 IF FLAG<>0 THEN GO SUB 780
770 GO TO 70
780 REM ** RESPOSTA **
790 PRINT ''   VALUE IS...''
800 IF B$<>''( )'' THEN PRINT''   '';B$
810 IF B$=''( )'' THEN PRINT''   NIL''
820 RETURN
830 REM ***********************
840 REM ** CAR **
850 IF S=2 THEN LET B$=A$(Z(2)+1 TO X(2))
860 IF S>2 THEN LET B$=A$(CFIRST TO CSECND)
870 RETURN
880 REM ***********************
890 REM ** CDR **
900 GO SUB 840
910 LET LB=LEN B$+7
920 LET B$=''(''+A$(LB TO EDGE+1)
930 IF B$(LEN B$-1 TO LEN B$)=''))''THEN LET B$=B$(TO LEN B$-1)
940 IF B$(2)='' ''THEN LET B$=''(''+B$(3 TO)
950 RETURN
960 REM ***********************
970 REM ** CONS **
980 LET B$=A$(7 TO LEN A$-1)
990 LET J=0
1000 IF B$(1)=''('' THEN LET J=1
1010 LET J=J+1
1020 IF B$(J)=''('' THEN GO TO 1050
1030 IF J<LEN B$ THEN GO TO 1010
1040 LET B$=''   >CONS ERROR<'':RETURN
1050 LET LB=LEN B$-1
1060 LET B$=''(''+B$( TO J-1)+B$(J+1 TO)
```

```
1070 LET B$=B$( TO LB+1)
1080 IF B$(LEN B$-1 TO )='' )'' THEN LET B$=B$(TO LEN B$-
     2)+'')''
1090 RETURN
1100 ***********************
1110 REM ** ATOM **
1120 LET A$=A$(7 TO LEN A$-1)
1130 LET J=0
1140 LET J=J+1
1150 IF A$(J)='' '' OR A$(J)=''('' THEN RETURN
1160 IF J<LEN A$ THEN GO TO 1140
1170 B$=''T''
1180 RETURN
1190 REM ***********************
1200 REM ** EQ **
1210 LET E=5: GO SUB 40
1220 LET J=0
1230 LET J=J+1
1240 IF A$(J)='')'' THEN RETURN
1250 IF A$(J)='' '' THEN GO TO 1280
1260 IF J<LEN A$ THEN GO TO 1230
1270 RETURN
1280 LET C$=A$(TO J-1)
1290 LET A$=A$(J+1 TO)
1300 IF C$=A$ THEN LET B$=''T''
1310 RETURN
1320 REM ***********************
1330 REM ** NULL **
1340 IF A$=''NULL ( )'' THEN LET B$='' * ILLEGAL - NULL NEEDS
     ARGUMENT *''
1350 IF A$=''NULL (())'' THEN LET B$=''T''
1360 RETURN
1370 REM ***********************
1380 REM ** MEMBRO **
1390 LET C$=A$(9 TO)
1400 LET J=1
1410 LET J=J+1
1420 IF C$(J)='')'' OR C$(J)=''(''THEN LET D$=C$ (TO J): GO
     TO 1450
1430 IF J<LEN(C$) THEN GO TO 1410
1440 RETURN
1450 LET J=LEN D$
1460 LET J=J+1
1470 IF C$(J TO J+LEN D$-1)=D$ THEN LET C$=C$( TO LEN C$-1): GO
     TO 1630
```

```
1480 IF J<LEN C$-LEN D$ THEN GO TO 1460
1490 RETURN
1500 REM ***********************
1510 REM ** MEMQ **
1520 LET C$=A$(7 TO )
1530 LET J=0
1540 LET J=J+1
1550 IF C$(J)='' '' THEN GO TO 1580
1560 IF J<LEN A$ THEN GO TO 1540
1570 RETURN
1580 LET D$=C$(TO J)
1590 LET C$=C$(J+2 TO)
1600 LET C$=C$ (TO LEN C$ -2)+'' ''
1610 LET J=0
1620 LET J=J+1
1630 IF C$(J TO J+LEN D$-1)=D$ THEN LET B$=''(''+'C$(J TO ):
     GO TO 1660
1640 IF J<LEN C$-LEN D$ THEN GO TO 1620
1650 RETURN
1660 LET B$=B$(TO LEN B$-1)+'')''
1670 IF B$(LEN B$-2 TO )=''))'' THEN LET B$=B$ ( TO LEN B$-1):
     GO TO 1670
1680 RETURN
1690 REM ***********************
1700 REM ** ACRESCENTAR **
1710 LET B$=A$(9 TO)
1720 LET B$=B$( TO Y(1)-9+'' ''+B$(Z(3)-7 TO )
1730 LET B$=B$( TO LEN B$ -1)
1740 RETURN
1750 REM **********************
1760 REM ** INVERTER **
1770 LET B$='' ''
1780 LET A$=A$(11 TO ): LET A$=A$(TO LEN A$-2)
1790 LET CT=0
1800 LET J=0
1810 LET J=J+1:IF J<LEN A$ THEN GO TO 1920
1820 IF A$(J)='' '' THEN GO TO 1850
1830 IF A$(J)=''('' THEN GO TO 1860
1840 GO TO 1810
1850 LET CT=CT+1: LET G$(CT)=A$(TO J-1): LET A$=A$(J+1 TO): GO
     TO 1800
1860 LET J=J+1: IF A$(J TO J+1)=''))'' THEN GO TO 1980
1870 IF A$(J)='')'' THEN GO TO 1910
1880 IF J=LEN A$ THEN GO TO 1900
1890 GO TO 1860
```

```
1900 LET CT=CT+1: LET G$(CT)=A$+'')'': GO TO 1930
1910 LET CT=CT+1: LET G$(CT)=A$(TO J): GO TO 1800
1920 LET CT=CT+1: LET G$(CT)=A$
1930 FOR M=CT TO 1 STEP -1
1940 LET B$=B$+G$(M): IF M>1 THEN LET B$=B$+'' ''
1950 NEXT M
1960 LET B$=''(''+B$+'')''
1970 RETURN
1980 LET CT=CT+1: LET G$(CT)=A$(TO J+1): LET A$=A$(J+2 TO ):
     GO TO 1800
1990 REM **********************
2000 REM ** EQUAL **
2010 LET E=8: GO SUB 40
2020 LET M=CODE A$: IF M>47 AND M<58 THEN GO TO 2370
2030 LET J=0
2040 LET J=J+1
2050 IF A$(J TO J+1)='') '' THEN LET J=J+1: GO TO 1280
2060 IF A$(J TO J+2)=''))) '' THEN GO TO 2100
2070 IF A$(J TO J+1)='')) '' THEN GO TO 2110
2080 IF J<LEN A$ THEN GO TO 2040
2090 RETURN
2100 LET C$=A$( TO J+2): LET A$=A$(J+4 TO ): GO TO 1300
2110 LET C$=A$( TO J+1): LET A$=A$(J+3 TO ): GO TO 1300
2120 REM ***********************
2130 REM ** LISTAR **
2140 LET E=7: GO SUB 40
2150 LET B$=''(''+A$+='')''
2160 RETURN
2170 REM ***********************
2180 REM ** SOMAR1 **
2190 LET E=7: GO SUB 40
2200 LET B$=STR$ (VAL A$+1)
2210 RETURN
2220 REM ***********************
2230 REM ** TIRAR1 **
2240 LET E=7: GO SUB 40
2250 LET B$=STR$ (VAL A$ -1)
2260 RETURN
2270 REM **********************
2280 REM ** ZEROP **
2290 IF FLAG=16 THEN LET E=8
2300 IF FLAG=31 THEN LET E=7
2310 GO SUB 40
2320 IF A$=''0'' AND FLAG=16 OR A$=''1'' AND FLAG=31 THEN LET
     B$=''T''
```

```
2330 RETURN
2340 REM ***********************
2350 REM ** DOIS ARGUMENTOS **
2360 LET E=13: GO SUB 40
2370 LET J=0
2380 LET J=J+1
2390 IF A$(J)='' '' THEN GO TO 2420
2400 IF J<LEN A$ THEN GO TO 2380
2410 LET B$='' * ERROR - ONLY ONE ARGUMENT *'': RETURN
2420 LET P=VAL A$(TO J-1))
2430 Q=VAL A$(J+1 TO )
2440 IF FLAG=17 THEN LET B$=STR$ (P-Q): RETURN
2450 IF FLAG=23 OR FLAG=25 THEN LET B=P/Q
2460 IF FLAG=25 THEN LET B=INT(.5+Q*(B-INT B))*1000))/1000
2470 IF FLAG=18 THEN LET B=P↑Q
2480 IF FLAG=11 AND P=Q THEN LET B$=''T''
2490 IF FLAG=27 AND P>Q THEN B$=''T''
2500 IF FLAG=28 AND P<Q THEN LET B$=''T''
2510 IF FLAG=32 THEN LET B=P-Q
2520 IF FLAG=11 OR FLAG>26 THEN RETURN
2530 LET B$=STR$ (B)
2540 RETURN
2550 REM **********************
2560 REM ** EXPT **
2570 LET E=7: GO SUB 40
2580 GO TO 2370
2590 REM ***********************
2600 REM ** MAX **MIN PLUS TIMES **
2610 LET F$=A$( TO 3): LET A$=A$(6 TO )
2620 LET CT=0: LET FLAG=0
2630 IF F$=''TIMES'' THEN LET CT=1
2640 LET J=0
2650 LET J=J+1
2660 IF A$(J)='' '' THEN GO TO 2690
2670 IF J<LEN A$ THEN GO TO 2650
2680 IF J=LEN A$ THEN LET FLAG=1
2690 LET P=VAL A$( TO J-1): IF FLAG=0 THEN LET A$=A$(J+1 TO)
2700 IF F$<>''PLUS'' AND CT=0 THEN CT=P
2710 IF F$=''MAX'' AND P>CT THEN LET CT=P
2720 IF F$=''MIN'' AND P<CT THEN LET CT=P
2730 IF F$=''PLUS'' THEN LET CT=CT+P
2740 IF F$=''TIMES'' THEN LET CT=CT*P
2750 IF FLAG=0 THEN GO TO 2640
2760 LET B$=STR$ (CT)
2770 RETURN
```

```
2780 REM ***********************
2790 REM ** MIN **
2800 GO TO 2610
2810 REM ***********************
2820 REM ** PLUS **
2830 LET F$=''PLUS''
2840 LET A$=A$(7 TO )
2850 GO TO 2620
2860 REM ***********************
2870 REM ** MINUS **
2880 LET E=8: GO SUB 40
2890 LET B$=STR$ (-VAL A$)
2900 RETURN
2910 REM ***********************
2920 REM ** QUOTIENT **
2930 LET E=11: GO SUB 40
2940 GO TO 2370
2950 REM ***********************
2960 REM ** RECIP **
2970 LET E=8: GO SUB 40
2980 IF A$=''0'' THEN LET B$='' DIVISION BY ZERO ILLEGAL'':
     RETURN
2990 LET B=1/VAL A$
3000 LET B$=STR$ B
3010 RETURN
3020 REM ***********************
3030 REM ** REMAINDER **
3040 LET E=12: GO SUB 40
3050 GO TO 2370
3060 REM **********************
3070 REM ** TIMES **
3080 LET F$=''TIMES''
3090 LET A$=A$(8 TO )
3100 GO TO 2620
3110 REM ***********************
3120 REM ** GREATERP **
3130 LET E=11: GO SUB 40
3140 GO TO 2370
3150 REM ***********************
3160 REM ** LESSP **
3170 LET E=8: GO SUB 40
3180 GO TO 2370
3190 REM ***********************
3200 REM **MINUSP **
3210 LET E=9: GO SUB 40
```

```
3220 IF VAL A$<0 THEN LET B$=''T''
3230 RETURN
3240 REM ***********************
3250 REM ** NUMBERP **
3260 LET A$=A$(10 TO )
3270 IF CODE A$>44 AND CODE A$<58 THEN LET B$=''T''
3280 RETURN
3290 REM ***********************
3300 REM ** DEFINE **
3310 LET A$=A$(9 TO)
3320 LET F$=A$( TO X(2)-9)
3330 LET G$=A$(X(4)-6 TO )
3340 LET J=0
3350 LET J=J+1
3360 IF G$(J)='' '' THEN GO TO 3390
3370 IF J<LEN G$ THEN GO TO 3350
3380 LET B$='' DEFINE ERROR'': RETURN
3390 LET G$=G$( TO J-1)
3400 LET NWDS=NWDS+1
3410 LET O$(NWDS)=G$: LET N$(NWDS)=F$
3420 LET B$=F$
3430 RETURN
3440 REM ***********************
3450 REM INICIALIZAR
3452 POKE 23658,8
3454 POKE 23609,40
3456 POKE 23692,255
3460 BORDER 1: PAPER 1: INK 7: CLS
3470 DIM G$(20,20): DIM O$(20,10): DIM N$(20,10): DIM X(12):
     DIM Y(12): DIM Z(12)
3480 LET NWDS=0: REM CONTAGEM DE PALAVRAS NOVAS DEFINIDAS
     PELO UTILIZADOR
3490 GO TO 70
```

MECÂNICO

```
10  REM MECÂNICO
20  MODE 6:VDU 19,0,4;0;
30  PRINT ''TEM PORTANTO PROBLEMAS COM O AUTOMÓVEL.''
40  PRINT ''VEJAMOS SE É POSSÍVEL RESOLVÊ-LOS...''
50  PRINT
60  PRINT ''ESCREVA UMA LETRA QUE DESCREVA O SEU PROBLEMA:''
70  PRINT
80  PRINT'' A - O MOTOR DE ARRANQUE NÃO FUNCIONA''
90  PRINT'' B - FUNCIONA MAS NÃO ARRANCA O MOTOR PRINCIPAL''
100 PRINT'' C - O MOTOR ARRANCA, MAS VAI ABAIXO
    IMEDIATAMENTE''
110 PRINT'' D - O CARRO ANDA MAS VAI MUITO ABAIXO''
120 PRINT'' E - O PONTO MORTO É IRREGULAR''
130 IF INKEY(0)<>'' '' THEN 130
140 A$=INKEY$(0)
150 IF A$<''A'' OR A$>''E'' THEN 140
160 GOSUB 1570
170 ON ASC(A$)-64) GOTO 200,580,1200,1230,1350
180 END
190 REM *************************
200 REM ARRANQUE NÃO FUNCIONA
210 PRINT''COMEÇAREMOS POR VERIFICAR A BATERIA.''
220 PRINT''LIGUE AS LUZES.''
230 PRINTTAB(6);''ESTÃO FRACAS?''
240 GOSUB 1520
250 IF A$=''N'' THEN 350:REM BATERIA OK
260 PRINT''OS CABOS DA BATERIA ESTÃO MAL FIXOS OU
    CORROÍDOS?''
270 GOSUB 1520
280 IF A$=''S'' THEN PRINT''FIXE-OS E LIMPE-OS.''
290 GOSUB 1590
300 PRINT''A CORREIA DA BATERIA ESTÁ MAL FIXA?''
```

```
310 GOSUB 1520
320 IF A$=''S'' THEN PRINTTAB(8);''APERTE A CORREIA DA
    VENTOINHA''
330 GOSUB 1590
340 PRINT''A LIGAÇÃO DIRECTA OU UM EMPURRÃO DEVEM BASTAR PARA
    ARRANCAR O MOTOR'':END
350 PRINT''O CABO DA BATERIA DO MOTOR DE ARRANQUE ESTÁ MAL
    FIXO OU CORROÍDO JUNTO A ESTE?''
370 GOSUB 1520
380 IF A$=''S'' THEN PRINT ''APERTE AS LIGAÇÕES E LIMPE-AS''
390 GOSUB 1590
400 PRINT''COLOQUE UMA PEÇA METÁLICA ENTRE OS TER-''
410 PRINT''MINAIS DO SOLENÓIDE. O MOTOR DE ARRANQUE
    FUNCIONA?''
420 GOSUB 1520
430 IF A$=''S'' THEN PRINT''O INTERRUPTOR DE IGNIÇÃO ESTÁ
    PROVAVELMENTE DEFEITUOSO''
440 GOSUB 1590
450 IF A$=''S'' THEN PRINT TAB(4);''DEVE SER
    SUBSTITUÍDO'':END
460 PRINT''O MOTOR DE ARRANQUE PARECE QUERER FUNCIONAR?''
470 GOSUB 1520
480 IF A$=''S'' THEN PRINT ''O MOTOR DE ARRANQUE PODE ESTAR
    PRESO'':GOTO 520
490 PRINT''COMO NADA ACONTECEU, O SOLENÓIDE ESTÁ''
500 PRINT''PROVAVELMENTE DEFEITUOSO. UM EMPURRÃO''
510 PRINT''PODERÁ ARRANCAR O AUTOMÓVEL.'':END
520 REM MOTOR DE ARRANQUE PRESO
530 PRINT''DESLIGUE A IGNIÇÃO, E ENGRENE A QUARTA''
540 PRINT''VELOCIDADE. EMPURRE UM POUCO O CARRO A''
550 PRINT''FIM DE LIBERTAR O MOTOR DE ARRANQUE.'':END
560 RETURN
570 REM *******************
580 REM MOTOR DE ARRANQUE FUNCIONA, MAS NÃO ARRANCA
590 PRINT''VERIFIQUE SE HÁ HUMIDADE NAS EXTREMIDADES''
600 PRINT''DOS CABOS. EXISTIRÃO ÁREAS HÚMIDAS?''
610 GOSUB 1520
620 IF A$=''S'' THEN PRINT''UTILIZE UM PULVERIZADOR CONTRA A
    HUMIDADE'':GOSUB 1590
630 PRINT''É VISÍVEL POEIRA NA PARTE ISOLANTE DA''
640 PRINT''RESISTÊNCIA, OU NA TAMPA DO DISTRIBUI-''
650 PRINT''DOR?''
660 GOSUB 1520
670 IF A$=''S'' THEN PRINT''LIMPE O INTERIOR E O EXTERIOR DO
    DISTRI-''
```

```
680 IF A$=''S'' THEN PRINT''BUIDOR.'':GOSUB 1590
690 PRINT''VERIFIQUE SE TODOS OS CABOS ESTÃO SECOS ANTES DE
    CONTINUAR''
700 GOSUB 1590
710 PRINT ''SE AINDA NÃO ARRANCA, DEVE VERIFICAR A''
720 PRINT''FAÍSCA DAS VELAS:'':GOSUB 1590
730 PRINT''A FAÍSCA SALTA UMA DISTÂNCIA GRANDE DEMAIS?''
740 GOSUB 1520
750 IF A$=''S'' THEN PRINT ''AS VELAS ESTÃO DEFEITUOSAS'':
    GOSUB 1590
760 IF A$=''N'' THEN 830
770 PRINT''AS VELAS ESTÃO OLEOSAS?''
780 GOSUB 1520
790 IF A$=''S'' THEN PRINT ''NÃO É POSSÍVEL FAZER UMA
    REPARAÇÃO DE''
800 IF A$=''S'' THEN PRINT''EMERGÊNCIA ENQUANTO AS VELAS
    ESTIVEREM NESSE ESTADO.'':GOSUB 1580:GOTO 770
810 IF A$=''N'' THEN PRINT''SE SE TRATA DE UMA EMERGÊNCIA,
    TENTE DI-''
820 IF A$=''N'' THEN PRINT''MINUIR A FOLGA PARA CERCA DE
    METADE DO NORMAL'':GOSUB 1590:GOTO 910
830 PRINT ''HÁ FAÍSCA?''
840 GOSUB 1520
850 IF A$=''S'' THEN 770
860 GOSUB 870:END
870 PRINT''VERIFIQUE SE HÁ RACHAS NO ROTOR,''
880 PRINT''ENROLAMENTO E TAMPA DO DISTRIBUIDOR.''
890 PRINT''SE NÃO HOUVER, O PROBLEMA DEVE RESIDIR NOS
    PLATINADOS OU CONDENSADOR''
900 PRINT ''PODERÁ SER NECESSÁRIO IR A UMA OFICINA'': RETURN
910 PRINT''TEM GASOLINA NO DEPÓSITO?''
920 GOSUB 1520
930 IF A$=''N'' THEN PRINT''ENCHA O DEPÓSITO E TENTE DE
    NOVO'':GOSUB 1590
940 PRINT''OS TUBOS DE GASOLINA E AR ESTÃO BEM FIXOS?''
950 GOSUB 1520
960 IF A$=''N'' THEN PRINT''FIXE-OS E TENTE DE NOVO'':GOSUB 1590
970 PRINT''REMOVA O FILTRO DE AR DO CARBURADOR''
980 GOSUB 1590
990 PRINT''PARECE SECO?''
1000 GOSUB 1520
1010 IF A$=''N'' THEN 1080
1020 PRINT''RODE UM POUCO O MOTOR TAPANDO COM A MÃO''
1030 PRINT''A ENTRADA DE AR'':GOSUB 1590
1040 PRINT''A MÃO ESTÁ HÚMIDA DE GASOLINA?''
```

```
1050 GOSUB 1520
1060 IF A$=''N'' THEN PRINT''DESAPERTE A TAMPA DA GASOLINA NO
     CASO DE''
1070 IF A$=''N'' THEN PRINT ''A ENTRADA DE AR ESTAR VEDADA. A
     BOMBA DE GASOLINA PODE NÃO ESTAR A TRABALHAR'':GOSUB 900
1080 PRINT''TENTOU MUITAS VEZES LIGAR O MOTOR NOS''
1090 PRINT''ÚLTIMOS MINUTOS?''
1100 GOSUB 1520
1110 IF A$=''N'' THEN 1150
1120 PRINT''ESPERE CERCA DE UM MINUTO E CARREGUE NO''
1130 PRINT''ACELERADOR ATÉ AO FUNDO SEM O BOMBEAR''
1140 PRINT''DEVE CONSEGUIR ARRANCAR'':END
1150 PRINT''O MOTOR PODE ESTAR AFOGADO E A VÁLVULA''
1160 GOSUB''FIXA NA POSIÇÃO ABERTA.'':GOSUB 1590
1170 PRINT''BATA NO LADO DO CARBURADOR E TENTE''
1180 PRINT''DE NOVO ARRANCAR'': STOP
1190 REM ********************
1200 REM ARRANCA, VAI ABAIXO
1210 PRINT''ISTO SUGERE UMA RESISTÊNCIA DEFEITUOSA QUE DEVE
     SER SUBSTITUIDA'':STOP
1220 REM ********************
1230 REM FUNCIONA, FALHA MUITO
1240 PRINT''O PROBLEMA É CAUSADO POR:''
1250 PRINTTAB(7);''CABOS EM CURTO-CIRCUITO OU MAL FIXOS;''
1260 PRINTTAB(7);''UMA FAÍSCA FRACA; OU''
1270 PRINTTAB(7);''UMA FALHA NO SISTEMA DE COMBUSTÍVEL''
1280 PRINT''VERIFIQUE PRIMEIRO OS CABOS'':GOSUB 1590
1290 PRINT''SE NÃO ESTIVEREM BEM, REPARE-OS. SE ESTIVEREM,
     PODERÁ TRATAR-SE DAS VELAS''
1300 GOSUB 870
1310 PRINT''PODE AINDA VERIFICAR AS VELAS''
1320 GOSUB 1590
1330 GOTO 1360
1340 REM ********************
1350 REM FUNCIONA, IRREGULAR EM PONTO MORTO
1360 PRINT''PODE ACONTECER QUE UMA OU MAIS VELAS''
1370 PRINT''ESTEJAM DEFEITUOSAS.'':GOSUB 1590
1380 PRINT''DESLIGUE-AS UMA DE CADA VEZ. AS QUE''
1390 PRINT''NÃO PROVOCAREM A PARAGEM DO MOTOR''
1400 PRINT''ESTÃO DEFEITUOSAS. VERIFIQUE-AS E VOLTE''
1410 PRINT''AO SISTEMA.'':GOSUB 1590
1420 PRINT''DESCOBRIU VELAS DEFEITUOSAS?''
1430 GOSUB 1520
1440 IF A$=''S'' THEN PRINT''SUBSTITUA TODAS AS VELAS, SE
     PUDER, OU''
```

```
1450 IF A$=''S'' THEN PRINT''APENAS AS DEFEITUOSAS'':GOTO
     1590
1460 PRINT''A CAUSA MAIS COMUM DE UM MAU PONTO''
1470 PRINT''MORTO, SE AS VELAS ESTÃO BOAS, É UMA''
1480 PRINT''MISTURA DE COMBUSTÍVEL DEMASIADO RICA.''
1490 PRINT''DEVE PORTANTO AJUSTÁ-LA.'':END
1500 PRINT''AJUSTE O JIGLER NO CARBURADOR'':END
1510 REM ********************
1520 PRINTTAB(16);(S - SIM, N - NÃO)?''
1530 IF INKEY$(0)<>'' '' THEN 1530
1540 A$=INKEY$(0)
1550 SOUND 16,-RND(15),RND(256)-1,5
1560 IF A$<>''S'' AND A$<>''N'' THEN 1540
1570 PRINTTAB(22);''>  OK    '';A$;'' <''
1580 VDU 7
1590 FOR T=1 TO 2000:NEXT T
1600 PRINT
1610 RETURN
```

MEDICI

```
10  REM MEDICI
20  MODE 6:VDU 19,0,4;0;
30  GOSUB 1250
40  PRINT''A - SOU BASTANTE GORDO''
50  PRINT''B - SOU UM POUCO GORDO''
60  PRINT''C - SOU LIGEIRAMENTE FORTE''
70  PRINT''D - TENHO O PESO CERTO''
80  PRINT''E - SOU MAIS MAGRO DO QUE DEVIA''
90  GOSUB 1170
100 WEIGHT=5*(ASC(A$)-68):IF A$=''E'' THEN WEIGHT=0
110 PRINT WEIGHT
120 GOSUB 1250
130 PRINT''FAÇO EXERCÍCIO, ELEVANDO O RITMO CARDÍA-''
140 PRINT''CO PARA 120 OU MAIS, PELO MENOS O SE-''
150 PRINT''GUINTE NÚMERO DE HORAS POR SEMANA:''
170 PRINT
180 PRINT''A - MENOS DE UM QUARTO DE HORA''
190 PRINT''B - ENTRE DE UM QUARTO E TRÊS QUARTOS DE HORA''
200 PRINT''C - ENTRE TRÊS QUARTOS DE HORA E HORA E MEIA''
210 PRINT''D - ENTRE UMA HORA E MEIA E DUAS HORAS E MEIA''
220 PRINT''E - MAIS DE DUAS HORAS E MEIA''
230 GOSUB 1170
```

```
240 EXERCISE=5*(ASC(A$)-63)-5:IF A$=''A'' THEN EXERCISE=0
250 PRINT EXERCISE
260 GOSUB 1250
270 PRINT''QUANDO GUIO:'':PRINT
280 PRINT''A - QUASE NUNCA USO O CINTO DE SEGURANÇA''
290 PRINT''B - USO O CINTO CERCA DE UM QUARTO DAS VEZES''
300 PRINT''C - USO O CINTO UMA DE CADA DUAS VEZES''
310 PRINT''D - USO O CINTO QUASE SEMPRE, MAS NEM SEMPRE''
320 PRINT''E - USO SEMPRE O CINTO DE SEGURANÇA''
330 GOSUB 1170
340 SEATBELT=2*ASC(A$)-65)
350 PRINT SEATBELT
360 GOSUB 1250
370 PRINT ''CONHEÇO OS PROBLEMAS DE NUTRIÇÃO E TENTO COMER DE
    FORMA SALUTAR:''
380 PRINT
390 PRINT''A - SEMPRE''
400 PRINT''B - QUASE SEMPRE''
410 PRINT''C - BASTANTES VEZES''
420 PRINT''D - DE VEZ EM QUANDO''
430 PRINT''E - QUASE NUNCA''
440 GOSUB 1170
450 DIET=-ASC(A$)+69
460 PRINT DIET
470 GOSUB 1250
480 PRINT''FUMO (UM CHARUTO CONTA COMO UM CIGARRO):''
490 PRINT
500 PRINT''A - NADA''
510 PRINT''B - MENOS DE QUINZE CIGARROS POR DIA''
520 PRINT''C - ENTRE 15 E 25 CIGARROS POR DIA''
530 PRINT''D - 26 A 42 CIGARROS POR DIA''
540 PRINT''E - MAIS DE 42 CIGARROS POR DIA''
550 GOSUB 1170
560 SMOKING=-7*(ASC(A$)-65)
570 PRINT SMOKING
580 GOSUB 1250
590 PRINT ''ALCOOL - QUANTAS BEBIDAS (EM MÉDIA) POR DIA?''
600 PRINT
610 PRINT''A - NENHUMA''
620 PRINT''B - MENOS DE 3''
630 PRINT''C - 3 A 6''
640 PRINT''D - 7 A 9''
650 PRINT''E - MAIS DE 9''
660 GOSUB 1170
670 DRINK=-30
```

```
680 IF A$=''A'' THEN DRINK=0
690 IF A$=''B'' THEN DRINK=1
700 IF A$=''C'' THEN DRINK=DRINK/5
710 IF A$=''D'' THEN DRINK=DRINK/2
720 PRINT DRINK
730 GOSUB 1250
740 PRINT''EM GERAL, QUE STRESS SENTIU NOS ÚLTIMOS''
750 PRINT''SEIS MESES?''
760 PRINT
770 PRINT''A - UM STRESS EXTREMO''
780 PRINT''B - UM STRESS RELATIVO''
790 PRINT''C - ALGUM STRESS''
800 PRINT''D - NEUTRO''
810 PRINT''E - SEM STRESS''
820 GOSUB 1170
830 STRESS=INT(2.5*(ASC(A$)-69))
840 PRINT STRESS
850 GOSUB 1300:CLS
860 PRINT''OPINIÃO PESSOAL DE MEDICI:''
870 PRINT
880 PRINT TAB(8);''PESO:'';WEIGHT
890 PRINT TAB(6);''EXERCÍCIO: '';EXERCISE
900 PRINT TAB(4);''SEGURANÇA DA CONDUÇÃO:'';SEATBELT
910 PRINT TAB(5);''NUTRIÇÃO:'';DIET
920 PRINT TAB(7);''FUMO:'';SMOKING
930 PRINT TAB(7);''ÁLCOOL:'';DRINK
940 PRINT TAB(8);''STRESS:''';STRESS
950 GOSUB 1300
960 ANT=WEIGHT+EXERCISE+SEATBELT+DIET+SMOKING+DRINK+STRESS
970 GOSUB 1300:PRINT
980 PRINT''   A SUA CLASSIFICAÇÃO GENÉRICA É'';ANT:PRINT
990 PRINT''   NUMA ESCALA EM QUE O ZERO É UM VALOR''
1000 PRINT''   MÉDIO, A MENOR CLASSIFICAÇÃO É INFE-''
1010 PRINT''RIOR A -80, E A MAIOR É SUPERIOR A 30''
1020 GOSUB 1300:PRINT
1030 IF ANT<6 AND ANT>-6 THEN A$=''MÉDIA'':
     L$=''62 A 73   72 A 78''
1040 IF ANT<-5 AND ANT>-21 THEN A$=''INFERIOR À MÉDIA'':
     L$=''60 A 66   65 A 71''
1050 IF ANT<-20 THEN A$=''FRACO'':
     L$=''60 OU MENOS   65 OU MENOS''
1060 IF ANT<-45 THEN A$='''MUITO FRACO''
1070 IF ANT<-60 THEN A$=''MUITO, MUITO FRACO''
1080 IF ANT>5 AND ANT< 15 THEN A$=''BOM'':
     L$=''74 A 80   79 A 85''
```

```
1090 IF ANT>14 THEN A$=''MUITO BOM'': L$=''81 +   86 +''
1100 PRINT ''ISTO INDICA QUE O SEU ESTADO DE SAÚDE É'';A$
1110 PRINT
1120 PRINT''EXPECTATIVA DE VIDA:''
1130 PRINT TAB(3);''HOMEM   MULHER''
1140 PRINT TAB(3);L$
1150 END
1160 REM ********************
1170 REM** ACEITAR ENTRADA
1180 IF INKEY$(0)<>'' ''THEN 1180
1190 A$=INKEY$(0)
1200 IF A$<''A'' OR A$>''E'' THEN 1190
1210 PRINT:PRINT TAB(12);''OK  '';A$
1220 RETURN
1230 REM ********************
1240 REM ATRASO/ESPAÇAR
1250 FOR J=1 TO 1000:NEXT J
1260 CLS
1270 PRINT:PRINT:PRINT:PRINT
1280 PRINT''QUAL DAS SEGUINTES OPÇÕES ESTÁ MAIS PRÓXIMA DA
     VERDADE:''
1290 PRINT
1300 FOR J=1 TO 400:NEXT J
1310 RETURN
```

RITA

```
10  REM RITA
20  PROCinicializar
30  PROCopções_de_saída
40  PROCopções_discriminação
50  PROCfazer_pergunta
60  PROCtomar_decisão_e_actualizar
70  PRINT''<RETURN>PARA CONTINUAR'';
80  IF X$<>'' '' THEN INPUT I$:GOTO 50
90  PRINT''APRENDIZAGEM''
100 PRINT''OU QUALQUER OUTRA TECLA E <RETURN>'' PARA USAR
    RITA''
110 INPUT X$:GOTO 50
120 END
130 REM ********************
140 DEF PROCfazer_pergunta
150 CLS
160 PRINT''EIS OS TEMAS SOBRE OS QUAIS POSSO DISCRIMINAR''
```

```
170 PRINT
180 FOR J=1 TO TT
190 PRINT ''>   '';A$(J)
200 NEXT J
210 GOSUB 1500
220 IF X$='' '' THEN PRINT''PENSE NUM, E CARREGUE EM
    <RETURN>''
230 IF X$<>'' '' THEN PRINT''ESTOU PRONTA A DETERMINAR QUAL
    É''
240 IF X$='' '' THEN INPUT J$
250 ADD=.5
260 FOR J=1 TO DQ
270 ADD=ADD+ADD
280 GOSUB 1500
290 IF X$<>'' ''AND TT>2 THEN 390:REM VERIFICAR SE A
    PERGUNTA PODE SER OMITIDA
300 PRINT''ESCREVA UM NÚMERO ENTRE''
310 PRINT''1 (VERDADEIRO) E 0 (FALSO) ($ PARA TERMINAR)''
320 PRINT:PRINT E$(J);''
330 INPUT H$:IF H$=''$'' THEN PRINT:PRINT''OBRIGADO'':
    :PRINT:END
340 C(J)=VAL(H$)
350 C(J)=ADD*C(J)
360 NEXT J
370 ENDPROC
380 REM ********************
390 REM VERIFICAR SE A PERGUNTA PODE SER OMITIDA
400 JUMP=1
410 FOR W=1 TO TT
420 IF ABS(B(W,J)-B(1,J))>.7 THEN JUMP=0
430 NEXT W
440 IF JUMP=0 THEN 300
450 C(JUMP)=B(W,J)
460 GOTO 360
470 REM ********************
480 DEF PROCtomar_decisão_e_actualizar
490 FOR J=1 TO TT
500 D(J)=0:E(J)=0:F(J)=0
510 NEXT J
520 ADD=.5
530 FOR J=1 TO TT
540 ADD=ADD+ADD
550 FOR X=1 TO DQ
560 REM JOGAR COM VALORES DAS TRÊS LINHÀS SEGUINTES PARA
    OBTER OS RESULTADOS MAIS EFICAZES
```

```
570 IF C(X)=B(J,X) THEN D(J)=D(J)+1
580 IF ABS(C(X)-B(J,X))<.6*ADD THEN E(J)=E(J)+.4
590 IF ABS(C(X)-B(J,X))<1.2*ADD THEN F(J)=F(J)+.1
600 NEXT X
610 NEXT J
620 A1=1:A2=1:A3=1
630 F1=1:F2=1:F3=1
640 FOR J=1 TO TT
650 IF D(J)>F1 THEN F1=D(J):A1=J
660 IF E(J)>F2 THEN F2=E(J):A2=J
670 IF F(J)>F3 THEN F3=F(J):A3=J
680 NEXT J
690 REM * ANUNCIAR RESULTADO *
700 PRINT
710 CFLG=0
720 PRINT''O RESULTADO MAIS PROVÁVEL É'';A$(A1)
730 IF A2<>A1 THEN PRINT''O MAIS PROVÁVEL A SEGUIR É''
    ;A$'A2):CFLG=1
740 IF A3 <>A2 AND A3<>A1 THEN PRINT''O MAIS PROVÁVEL A
    SEGUIR É'';A$(A3):CFLG=2
750 PRINT
760 PRINT''O RESULTADO MAIS PROVÁVEL ESTÁ CORRECTO?
770 INPUT F$
780 IF F$<>''S'' AND F$<>''N'' THEN 770
790 IF F$<>''S'' AND X$<>'' ''THEN ENDPROC
800 IF F$=''S'' THEN 980
810 IF TT=2 AND A1=1 THEN A1=2:GOTO 980
820 IF TT=2 THEN A1=1:GOTO 980
830 IF CFLG=0 THEN 890
840 PRINT''A SEGUNDA OPÇÃO ESTÁ CORRECTA?
850 INPUT F$
860 IF F$=''N'' THEN 890
870 IF CFLG=1 THEN A1=A2:GOTO 980
880 IF CFLG=2 THEN A1=A3:GOTO 980
890 GOSUB 1500
900 FOR J=1 TO TT
910 PRINT J;''-  '';A$(J)
920 NEXT J
930 PRINT
940 PRINT''QUAL É A CORRECTA?''
950 INPUT A1
960 IF A1<1 OR A1>TT THEN 950
970 REM ** EDUCANDO RITA **
    (ACTUALIZAR BASE DE CONHECIMENTOS)
980 FOR J=1 TO DQ
```

```
990 IF B(A1,J)<>0 THEN B(A1,J)=(C(J+5*B (A1,J))/6)
1000 IF B(A1,J)=0 THEN B(A1,J)=C(J)
1010 B(A1,J)=INT(10*B(A1,J))/10
1020 NEXT J
1030 PRINT
1040 IF U$='' '' THEN ENDPROC
1050 FOR J=1 TO TT
1060 PRINT:GOSUB 1500
1070 PRINT A$(J)
1080 PRINT
1090 FOR K=1 TO DQ
1100 PRINT E$(K);'' '';B(J,K)
1110 NEXT K
1120 NEXT J
1130 PRINT
1140 ENDPROC
1150 REM ********************
1160 DEF PROCopções_de_saída
1170 TT=0
1180 TT=TT+1
1190 GOSUB 1500
1200 PRINT''ESCREVA NÚMERO DA OPÇÃO''TT''
       (<RETURN>PARA TERMINAR)''
1210 INPUT LINE A$(TT)
1220 IF A$(TT)='' '' OR TT=51 THEN TT=TT-1:ENDPROC
1230 GOTO 1180
1240 REM ********************
1250 DEF PROCopções_discriminação
1260 CLS
1270 FOR J=1 TO TT
1280 PRINT A$(J)
1290 NEXT J
1300 DQ=0
1310 DQ=DQ+1
1320 GOSUB 1500
1330 PRINT''ESCREVA NÚMERO DA PERGUNTA ''DQ'' (<RETURN>PARA
     TERMINAR)''
1340 INPUT E$(DQ)
1350 IF E$(DQ)='' '' OR DQ=51 THEN DQ=DQ-1:ENDPROC
1360 GOTO 1310
1370 REM ********************
1380 DEF PROCinicializar
1390 CLS
1400 REM REDUZIR ARRAYS NA LINHA SEGUINTE DE ACORDO COM
     NECESSIDADES
```

```
1410 DIM
     A$(50),B(50,50),C(50),D(50),E$(50),E(50),F( 50),G(50)
1420 X$='' ''
1430 PRINT''CARREGUE NUMA TECLA E EM <RETURN> SE''
1440 PRINT''QUISER OBSERVAR A BASE DE CONHECIMENTOS''
1450 PRINT''ACTUALIZADA DEPOIS DE CADA EXECUÇÃO; OU''
1460 PRINT''SÓ EM <RETURN> SE NÃO QUISER''
1470 INPUT U$
1480 CLS
1490 ENDPROC
1500 PRINT''------------------------------------
------''
1510 RETURN
```

ESCALAS

```
10  REM ESCALAS
20  DIM X(50),Z(50)
30  PRINT''
{CLR}''
40  INPUT ''MAIOR VALOR'';A
50  INPUT ''MENOR VALOR'';B
60  A=A+.001
70  B=B+.001
80  C=(A-B)/50
90  X(0)=B
100 FOR J=1 TO 50
110 X(J)=X(J-1)+C
120 Z(J)=J/50
130 PRINT Z(J),X(J)
140 NEXT J
150 DFF=(X(2)-X(1))/2
160 COUNT=0
170 COUNT=COUNT+1
180 PRINT''INDIQUE VALOR'';Q$
190 INPUT Q$
200 IF Q$='' ''THEN END
210 Q=VAL(Q$)
220 IF Q <B OR Q> A THEN 180
230 FOR J=1 TO 50
240 IF ABS(Q-X(J))<DEF THEN PRINT COUNT;''-'';Z(J)
250 NEXT J
260 GOTO 170
```

HASTE

```
10  REM HASTE
20  DIM A$(255),B$(255)
30  F='' '':F=0:CLS
40  REM ********************
50  FLAG=0
60  INPUT''>   ''D$
70  IF D$='' '' THEN 60
80  IF LEFT$(D$,1)=''?'' THEN 200
90  E=0
100 E=E+1
110 IF MID$(D$,E,1)=''*'' THEN 140
120 IF E<LEN(D$) THEN 100
130 PRINT ''ENTRADA INCORRECTA'':GOTO 50
140 IF FLAG=3 THEN RETURN
150 F=F+1:IF F=256 THEN END
160 A$(F)=LEFT$(D$,E-1)
170 B$(F)=MID$(D$,E+1)
180 GOTO 50
190 REM ********************
200 REM INTERROGAR
210 FLAG=4:true=0
220 IF RIGHT$(D$,1)=''/'' THEN FLAG=3
230 FOR J=1 TO LEN(D$)-5
240 IF MID$(D$,J,5)=''AND'' THEN FLAG=15:true=J
250 NEXT J
260 IF FLAG=5 THEN 410
270 IF LEFT$(D$,3)=''?/*'' THEN FLAG=1
280 IF LEFT$(D$,4)=''?/*/'' THEN FLAG=2
290 IF FLAG=3 THEN GOSUB 90:F$=MID$(D$,2,E-2)
300 IF FLAG=1 THEN F$=MID$(D$,4)
310 E=0
320 E=E+1
330 IF A$(E)='' '' AND FLAG=4 AND true=0 THEN PRINT ''FALSE''
340 IF A$(E)='' '' THEN 520
350 IF FLAG=4 AND ''?''+A$(E)+''*''+B$(E)=D$ THEN PRINT
    ''TRUE'':true=1
360 IF FLAG=3 AND F$=A$(E) THEN PRINT B$ (E)
370 IF FLAG=2 THEN PRINT A$(E);''*'';B$(E)
380 IF FLAG=1 AND F$=B$(E) THEN PRINT A$ (E)
390 IF E<255 THEN 320
400 REM ********************
410 F$=MID$(D$,4,true-4):G$=MID$(D$,true+7)
420 E=0
```

```
430 E=E+1
440 IF A$(E)='' '' THEN 520
450 IF B$(E)=F$ THEN 470
460 IF E<255 THEN 430
470 H=0
480 H=H+1
490 IF B$(H)='' '' THEN 460
500 IF B$(H)=G$ AND A$(E)=A$(H) THEN PRINT A$(E)
510 IF H<255 THEN 480
520 PRINT TAB(5);''>END OF ANSWER<''
530 GOTO 50
```

EASLE

```
10  REM EASLE
20  MODE 6:VDU 19,0,4;0;
30  DIM X(40),Y(40),Z(40)
40  PRINT
50  INPUT''> ''A
60  IF A$='' '' THEN END
70  REM ********************
80  FOR J=1 TO 40
90  X(J)=0:Y(J)=0:Z(J)=0
100 NEXT J
110 REM ********************
120 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
130 FOR J=1 TO LEN(A$)
140 B$=MID$(A$,J,1)
150 IF B$=''('' THEN S=S+1:Z(S)=J:IF T=0 THEN CFIRST=J
160 IF B$='')'' THEN T=T+1:Y(T)=J:IF CSECND <>0 AND EDGE=0
    THEN EDGE=J-1
170 IF T=1 AND B$='')'' THEN CSECND=J
180 IF B$= '' '' THEN R=R+1:X(R)=J
190 NEXT J
200 IF S=T THEN 260:REM EQUILÍBRIO ( )
210 IF S<T THEN PRINT'' ->MISSING (''
220 IF C>T THEN PRINT '' ->MISSING )''
230 INPUT ''>    '',B$:
240 A$=A$+B$
250 GOTO 80
260 FLAG=0
270 IF LEFT$(A$,5)=''CAR ('' THEN FLAG=1
280 IF LEFT$(A$,5)=''CDR ('' THEN FLAG=2
290 IF LEFT$(A$,6)=''CONS ('' THEN FLAG=3
```

```
300 IF LEFT$(A$,6)=''ATOM ('' THEN FLAG=4
310 IF LEFT$(A$,4)=''EQ ('' THEN FLAG=5
320 IF LEFT$(A$,6)=''NULL ('' THEN FLAG=6
330 ON FLAG GOSUB 420,470,550,690,780,920
340 IF FLAG<>0 THEN GOSUB 360
350 GOTO 40
360 REM ** RESPOSTA **
370 PRINT''   VALUE IS...''
380 IF B$<>''( )'' THEN PRINT''  '';B$
390 IF B$=''( )'' THEN PRINT''  NIL''
400 RETURN
410 REM ********************
420 REM ** CAR **
430 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z(2))
440 IF S>2 THEN B$=MID$(A$,CFIRST,CSECND-CFIRST+1)
450 RETURN
460 REM ********************
470 REM ** CDR **
480 GOSUB 420
490 LB=LEN(B$)+7
500 B$=''(''+MID$(A$,LB,EDGE-1)
510 IF RIGHT$(B$,2)=''))'' THEN B$=LEFT$(B$,LEN(B$)-1)
520 IF MID$(B$,2,1)='' '' THEN B$=''(''+MID$(B$,3)
530 RETURN
540 REM ********************
550 REM ** CONS **
560 B$=MID$(A$,7,LEN(A$)-1)
570 J=0
580 IF LEFT$(B$,1)=''('' THEN J=1
590 J=J+1
600 IF MID$(B$,J,1)=''('' THEN 630
610 IF J<LEN(B$) THEN 630
620 B$=''  >CONS ERROR<'':RETURN
630 LB =LEN(B$)-1
640 B$=''(''+LEFT$(B$,J-1)+MID$(B$,J+1)
650 B$=LEFT$(B$,LB)
660 IF RIGHT$(B$,2)='' )'' THEN B$=LEFT$(B$,LEN(B$)-2)+'')''
670 RETURN
680 REM ********************
690 REM ** ATOM **
700 A$=MID$(A$,7,LEN(A$)-1)
710 J=0:B$=''NIL''
720 J=J+1
730 IF MID$(A$,J,1)='' '' OR MID$(A$,J,1)=''('' THEN RETURN
740 IF J<LEN(A$) THEN 720
```

```
750 B$=''T''
760 RETURN
770 REM ********************
780 REM ** EQ **
790 A$=MID$(A$,5)
800 A$=LEFT(A$,LEN(A$)-1)
810 J=0:B$=''NIL''
820 J=J+1
830 IF MID$(A$,J,1)='')'' THEN RETURN
840 IF MID$(A$,J,1)='' '' THEN 870
850 IF J<LEN(A$) THEN 820
860 RETURN
870 C$=LEFT$(A$,J-1)
880 A$=MID$(A$,J+1)
890 IF C$=A$ THEN B$=''T''
900 RETURN
910 REM ********************
920 REM ** NULL **
930 B$=''NIL''
940 IF A$=''NULL ( )'' THEN PRINT''ILLEGAL - NULL NEEDS
    ARGUMENT'':B$='' ''
950 IF A$=''NULL (( ))'' THEN B$=''T''
960 RETURN
```

PROLOG-A

```
10  REM PROLOG-A (SIMPLE FRONT-END)
20  REM * TODAS AS ENTRADAS EM MAIÚSCULAS *
30  GOTO 50
40  PRINT ''NO (MORE) ANSWERS'':RETURN
50  MODE VDU 19,0,4;0;:PROCinicializar
60  REM ********************
70  PRINT
80  INPUT ''&.''J$
90  IF J$='' '' THEN END
100 PRINT IF J$=''LIST ALL'' THEN GOSUB 1860:GOTO 70
110 IF LEFT$(J$,5)=''LIST '' THEN J$=MID$(J$,5)+'' '':GOSUB
    990:GOTO 70
120 IF RIGHT$(J$,1)<>'')'' THEN PRINT''1.'': INPUT
    M$:J$+M$:GOTO 120
130 LJ=LEN(J$)
140 J$=LEFT$(J$,LJ-1)+'' '':REM ELIMINAR ) FINAL, SUBSTITUIR
    POR ESPAÇO
150 LJ=LEN(J$)
```

```
160 FLAG=0
170 IF LEFT$(J$,4)=''ADD(''THEN J$=MID$(J$,5):FLAG=1
180 RULEFLAG=0:PLUSFLAG=0:ARITHFLAG=0
190 FOR R=1 TO LEN(J$)
200 IF MID$(J$,R,4)='' IF '' THEN RULEFLAG=R:FLAG=6
210 IF MID$(J$,R,5)='' AND '' THEN PLUSFLAG=R
220 IF MID$(J$,R,4)=''SUM('' THEN ARITHFLAG=1
230 IF MID$(J$,R,6)=''TIMES('' THEN ARITHFLAG=2
240 IF MID$(J$,R,6)='' LESS '' THEN ARITHFLAG=3
250 IF MID$(J$,R,3)='' INT '' THEN ARITHFLAG=4
260 NEXT R
270 IF LEFT$(J$,3)=''IS('' THEN J$=MID$(J$,4):FLAG=2
280 IF LEFT$(J$,10)=''WHICH(X : '' THEN J$=MID$(J$,11):FLAG=3
290 IF LEFT$(J$,16)=''WHICH((X Y) . X '' THEN
    J$=MID$(J$,17):FLAG=4
300 IF FLAG=0 THEN PRINT''SYNTAX ERROR'':GOTO 70
310 LJ=LEN(J$)
320 REM ENVIAR PARA SUBROTINAS APROPRIADAS
330 IF PLUSFLAG<>0 THEN GOSUB 1950:GOTO 70:REM CODIFICAR
    REGRA CONTENDO AND
340 IF RULEFLAG<>0 AND FLAG<>5 THEN GOSUB 1110:REM
    CODIFICAR REGRA
350 IF ARITHFLAG<>0 THEN GOSUB 2430:GOTO 70:REM ARITMÉTICA
360 IF RIGHT$(J$,3)='' X '' OR RIGHT$(J$,3)='' S '' THEN
    J$=LEFT$(J$,LJ-2)+'' ''
370 LJ=LEN(J)
380 IF FLAG=1 THEN GOSUB 440:REM ADD
390 IF FLAG=2 THEN GOSUB 520:REM IS
400 IF FLAG=3 THEN GOSUB 610:REM WHICH
410 IF FLAG=4 THEN GOSUB 830:REM WHICH2
420 GOTO 70
430 REM ********************
440 REM    ADD
450 K=0
460 K=K+1
470 IF Z$(K)='' ''THEN Z$(K)=J$:RETURN
480 IF K<1000 THEN 460
490 PRINT''MEMORY FULL''
500 RETURN
510 REM ********************
520 REM    IS
530 K=0
540 K=K+1
550 IF Z$(K)='' '' THEN 580
560 IF Z$(K)=J$ THEN PRINT''YES'':GOTO 590
```

```
570 IF K<1000 THEN 540
580 PRINT''NO''
590 RETURN
600 REM ********************
610 REM       WHICH
620 IF LEFT$(J$,1)=''X'' THEN 710
630 J$=LEFT$(J$,LJ-1)
640 K=0
650 K=K+1
660 IF Z$(K)='' '' THEN 690
670 IF J$=LEFT$(Z$(K),LEN(J$)) THEN PRINT
    RIGHT$(Z$(K),(LEN(Z$(K))-LEN(J$)))
680 IF K<1000 THEN 650
690 GOSUB 40
700 RETURN
710 REM * PERGUNTA COMEÇANDO POR X*
720 J$=MID$(J$,3,LEN(J$)-3)
730 LJ=LEN(J$)
740 K=0
750 K=K+1
760 IF Z$(K)='' '' THEN 800
770 Q$=MID$(Z$(K),LEN(Z$(K))-LJ,LJ)
780 IF Q$=J$ THEN PRINT LEFT$(Z$(K),LEN(Z$(K))-LJ+2))
790 IF K<1000 THEN 750
800 GOSUB
810 RETURN
820 REM ********************
830 REM   WHICH2
840 J$=LEFT$(J$,LJ-2)
850 LJ=LEN(J$)
860 K=0
870 K=K+1
880 IF Z$(K)='' '' THEN 960
890 LFLAG=0
900 FOR L=1 TO LEN (Z$(K))-LJ
910 IF MID$(Z$(K),L,LJ)=J$ THEN LFLAG=L
920 NEXT L
930 IF LFLAG=0 THEN 950
940 PRINT LEFT$(Z$(K),LFLAG-2);MID$(Z$(K),LFLAG+LJ))
950 IF K<1000 THEN 870
960 GOSUB 40
970 RETURN
980 REM ********************
990 REM   LIST
1000 K=0
```

```
1010 K=K+1
1020 IF Z$(K)='' '' THEN RETURN
1030 LFLAG=0
1040 FOR L=1 TO LEN(Z$(K))-LEN(J$)
1050 IF MID$(Z$(K),L,LEN(J$))=J$)) THEN LFLAG=1
1060 NEXT L
1070 IF LFLAG=1 THEN PRINT Z$(K)
1080 IF K<1000 THEN 1010
1090 RETURN
1100 REM ********************
1110 REM   FORMAR REGRAS
1120 R=RULEFLAG
1130 E$=LEFT$(J$,R):F$=MID$(J$,R+4)
1140 IF LEFT$(E$,1)<>''X'' THEN PRINT''RULE ERROR'':GOTO 70
1150 REM A LINHA SEGUINTE DETECTA ENTRADAS DO TIPO X COME Y SE
     X É_UM Y
1160 IF RIGHT$(F$,2)=''S'' THEN 1390
1170 PRINT TAB(18);''COMPILING RULE''
1180 FOR T=1 TO 100
1190 R$(T)='' ''
1200 NEXT T
1210 E$=MID$(E$,3):F$=MID$(F$,3)
1220 K=0:RR=0
1230 K=K+1
1240 IF Z$(K)='' '' THEN 1300
1250 IF RIGHT$(Z$(K),LEN(F$))<>F$ THEN 1370
1260 RR=RR+1
1270 R$(RR)=LEFT$(Z$(K),(LEN(Z$(K))-LEN(F$)))+E$
1280 PRINT ''>  '';R$(RR)
1290 GOTO 1230
1300 IF RR=0 THEN RETURN
1310 RC=0
1320 RC=RC+1
1330 Z$(K)=R$(RC)
1340 IF K<1000 THEN K=K+1
1350 IF RC<RR THEN 1320
1360 RETURN
1370 IF K<1000 THEN 1230
1380 RETURN
1390 REM ** REGRA COM DUAS VARIÁVEIS *
1400 FOR T=1 TO 100
1410 R$(T)='' ''
1420 NEXT T
1430 K=0:RR=0
1440 IF K<1000 THEN RETURN
```

```
1450 K=K+1
1460 IF Z$(K)='' '' THEN 1770
1470 REM DIVIDIR EM TRÊS PALAVRAS
1480 Q$=Z$(K)
1490 J=0
1500 J=J+1
1510 IF MID$(Q$,J,1)='' '' THEN 1540
1520 IF J<LEN(Q$) THEN 1500
1530 PRINT''RULE COMPILING ERROR'':GOTO 70
1540 A$=LEFT(Q$,J)
1550 Q$=MID$(Q$,J+1)
1560 J=0
1570 J=J+1
1580 IF MID$(Q$,J,1)='' '' THEN 1610
1590 IF J<LEN(Q$) THEN 1570
1600 PRINT''RULE COMPILING ERROR'':GOTO 70
1610 B$=LEFT$(Q$,J)
1620 Q$=MID$(Q$,J+1)
1630 J=0
1640 J=J+1
1650 IF MID$(Q$,J,1)='' '' THEN 1680
1660 IF J<LEN(Q$) THEN 1640
1670 PRINT''RULE COMPILING ERROR'':GOTO 70
1680 PRINT TAB(18);''COMPILING RULE''
1690 C$=LEFT$(Q$,J)
1700 M$=MID$(F$,3,LEN(B$))
1710 IF B$<>M$ THEN 1440
1720 RR=RR+1
1730 N$=MID$(E$,3,LEN(E$)-4)
1740 R$(RR)=A$+N$+C$
1750 PRINT''>   '';R$(RR)
1760 GOTO 1440
1770 IF RR=0 THEN RETURN
1780 M=0
1790 M=M+1
1800 IF M>RR THEN RETURN
1810 Z$(K)=R$(M)
1820 IF K=1000 THEN PRINT ''OUT OF MEMORY'':GOTO 70
1830 K=K+1
1840 GOTO 1790
1850 REM ******************
1860 REM   LISTAR TUDO
1870 PRINT
1880 K=0
1890 K=K+1
```

```
1900 IF Z$(K)='' '' THEN RETURN
1910 PRINT Z$(K)
1920 IF K<1000 THEN 1890
1930 RETURN
1940 REM ********************
1950 REM FORMAR REGRAS COM ''AND'' DO SEGUINTE TIPO:
1960 REM (X COME Y SE X É_AVE E Y VEM EM CAIXAS)
1970 REM A DECLARAÇÃO X DEVE ESTAR ANTES DE Y EM TODOS OS CASOS
1980 REM DIVIDIR EM SECÇÕES
1990 J$=MID$(J$,2):REM TIRAR ''X''
2000 PRINT TAB(20);''COMPILING RULE''
2010 J=1
2020 J=J+1
2030 IF MID$(J$,J,1)='' '' THEN 2060
2040 IF J<LEN(J$) THEN 2020
2050 PRINT ''RULE COMPILING ERROR'':RETURN
2060 A$=LEFT$(J$,J):REM RELAÇÃO 1
2070 J$=MID$(J$,J+7):REM ELIMINAR ATÉ AO INÍCIO DA SEGUNDA
     RELAÇÃO
2080 J=1:count=0
2090 J=J+1
2100 IF MID$(J$,J,1)='' '' THEN count=count+1
2110 IF count=2 THEN 2140
2120 IF J<LEN(J$) THEN 2090
2130 PRINT''RULE COMPILING ERROR'':RETURN
2140 B$=LEFT$(J$,J):REM DECLARAÇÃO1
2150 C$=MID$(J$,J+6):REM DECLARAÇÃO2
2160 IF C$=CHR$(32) THEN PRINT''RULE COMPILING
     ERROR'':RETURN
2170 REM PERCORRER BASE DE DADOS
2180 FOR T=1 TO 200
2190 R$(T)='' ''
2200 NEXT T
2210 R1=0:R2=99
2220 K=0
2230 K=K+1
2240 IF Z$(K)='' '' THEN 2310
2250 IF R1=99 OR R2=200 THEN PRINT''MEMORY SHORTAGE'':GOTO
     2310
2260 LB=LEN(B$)
2270 IF RIGHT$(Z$(K),LB)=B$ THEN R1=R1+1:R$(R1)=LEFT$(Z$(K),
     LEN(Z$(K))-LB)
2280 LC=LEN(C$)
2290 IF RIGHT$(Z$(K),LC)=C$ THEN R2=R2+1:R$(R2)=LEFT$(Z$(K),
     LEN(Z$(K))-LC)
```

```
2300 IF K<1000 THEN 2230
2310 IF R$(100)='' '' THEN PRINT''STATEMENT 2 OF INPUT NOT IN
     DATA BASE'':RETURN
2320 REM CODIFICAR REGRAS
2330 R1=0:R2=99
2340 R2=R2+1
2350 R1=R1+1
2360 IF R$(R1)>'' '' AND R$(R2)>'' '' THEN
     Z$(K)=R$(R1)+A$+R$(R2)+'' ''
2370 PRINT''>  '';Z$(K)
2380 K=K+1
2390 IF R$(R2+1)<>'' '' THEN 2340
2400 IF R$(R1+1)<>'' '' THEN 2350
2410 RETURN
2420 REM ********************
2430 REM   ARITMÉTICA
2440 LJ=LEN(J$)
2450 IF ARITHFLAG<3 THEN GOSUB 2490
2460 IF ARITHFLAG=3 THEN GOSUB 2890
2470 IF ARITHFLAG=4 THEN GOSUB 3080
2480 RETURN
2490 REM *** SOMA ***
2500 J$=MID$(J$,5,LJ-5)
2510 IF LEFT$(J$,2)=''S('' THEN J$=MID$(J$,3)
2520 LJ=LEN(J$)
2530 K=0
2540 K=K+1
2550 IF MID$(J$,K,1)='' '' THEN A$=LEFT$(J$,K-
     1)J$=MID$(J$,K+1): GOTO 2580
2560 IF K<LJ THEN 2540
2570 PRINT TAB(12);''ARITHMETIC ERROR'';RETURN
2580 LJ=LEN(J$)
2590 K=0
2600 K=K+1
2610 IF MID$(J$,K,1)='' '' THEN B$=LEFT$(J$, J-
     1):J$=MID$(J$,K+1): GOTO 2460
2620 IF K<LJ THEN 2600
2630 PRINT TAB(12);''ARITHMETIC ERROR'':RETURN
2640 LJ=LEN(J$)
2650 K=0
2660 K=K+1
2670 IF MID$(J$,K,1)='')'' THEN C$=LEFT$(J$,K-1):GOTO 2700
2680 IF K<LJ THEN 2660
2690 PRINT TAB(12);''ARITHMETIC ERROR (TOO MANY
     VARIABLES)'':RETURN
```

```
2700 AN=0:BN=0:CN=0
2710 IF ASC(A$)>58 THEN AN=1
2720 IF ASC(B$)>58 THEN BN=2
2730 IF ASC(C$)>58 THEN CN=4
2740 GUIDE=AN+BN+CN:IF GUIDE=3 OR GUIDE=5 OR GUIDE=6
     THEN 2690
2750 IF ARITHFLAG=2 THEN 2820:REM TIMES
2760 IF GUIDE>0 THEN 2790
2770 IF VAL(A$)+VAL(B$)=VAL(C$) THEN PRINT''YES'':RETURN
2780 PRINT''NO'':RETURN
2790 IF GUIDE=1 THEN PRINT VAL(C$)-VAL(B$):GOSUB 40:RETURN
2800 IF GUIDE=2 THEN PRINT VAL(C$)-VAL(A$):GOSUB 40:RETURN
2810 PRINT VAL(A$)+VAL(B$):GOSUB 40:RETURN
2820 REM ** TIMES **
2830 IF GUIDE>0 THEN 2860
2840 IF VAL(A$)*VAL(B$)=VAL(C$) THEN PRINT''YES'':RETURN
2850 PRINT''NO'':RETURN
2860 IF GUIDE=1 THEN PRINT VAL(C$)/VAL(B$);GOSUB 40:RETURN
2870 IF GUIDE=2 THEN PRINT VAL(C$)/VAL(A$):GOSUB 40:RETURN
2880 PRINT VAL(A$)*VAL(B$):GOSUB 40:RETURN
2890 REM *************** LESS ***************
2900 NF=0
2910 IF ASC(J$)<58 THEN NF=1:REM NUMEROS
2920 count=0
2930 K=0
2940 K=K+1
2950 IF MID$(J$,K,1)='' '' THEN count=count+1
2960 IF count=2 THEN 3000
2970 IF K<LEN(J$) THEN 2940
2980 PRINT TAB(20);''COMPARISON ERROR''
2990 RETURN
3000 B$=MID$(J$,K+1)
3010 A$=LEFT$(J$,K-6)
3020 IF NF=1 THEN 3050
3030 IF A$<B$ THEN PRINT ''YES'':RETURN
3040 PRINT''NO'':RETURN
3050 REM * NÚMEROS *
3060 IF VAL(A$)<VAL(B$) THEN PRINT''YES'':RETURN
3070 PRINT''NO'':RETURN
3080 REM ********* INT **********
3090 IF RIGHT$(J$,2)=''X'' THEN 3190
3100 K=0
3110 K=K+1
3120 IF MID$(J$,K,1)='' '' THEN 3160
3130 IF K<LEN(J$) THEN 3110
```

```
3140 PRINT TAB(20);''ARITHMETIC ERROR''
3150 RETURN
3160 A=VAL(LEFT$(J$,K-1))
3170 IF INT(A)=A THEN PRINT''YES'':RETURN
3180 PRINT''NO'':RETURN
3190 K=0
3200 K=K+1
3210 IF MID$(J$,K,1)='' '' THEN 3240
3220 IF K<LEN(J$) THEN 3200
3230 PRINT TAB(20);''ARITHMETIC ERROR'':RETURN
3240 PRINT INT (VAL(LEFT$(J$,K-1)))
3250 RETURN
3260 REM ********************
3270 DEF PROCinicializar
3280 CLS
3290 DIM Z$(1000),R$(200)
3300 ENDPROC
```

SSLISP

```
10  REM S.S. LISP
20  MODE 6:VDU 19,0,4;0;:GOTO 3450:REM INICIALIZAR
30  REM ********************
40  A$=MID$(A$,E):A$=LEFT$(A$,LEN(A$)-1)
50  RETURN
60  REM ********************
70  PRINT
80  NN=0
90  A$='' '':INPUT'': ''A$
100 IF A$='' '' THEN END:REM CARREGAR EM <RETURN> PARA TER-
    MINAR
110 REM ********************
120 FOR J=1 TO 12
130 X(J)=0:Y(J)=0:Z(J)=0
140 NEXT J
150 REM ********************
160 R=0:S=0:T=0:CFIRST=0:CSECND=0:EDGE=0
170 FOR J=1 TO LEN(A$)
180 B$=MID$(A$,J,1)
190 IF B$=''(''THEN S=S+1:Z(S)=J:IF T=0 THEN CFIRST=J
200 IF B$='')'' THEN T=T+1:Y(T)=J:IF CSECN D<>0 AND EDGE=0
    THEN EDGE=J-1
210 IF T=1 AND B$='')'' THEN CSECND=J
```

```
220 IF B$='' '' THEN R=R+1:X(R)=J
230 NEXT J
240 IF S=T THEN 300:REM EQUILÍBRIO DE( )
250 IF S<T THEN PRINT '' ->MISSING (''
260 IF S>T THEN PRINT '' ->MISSING )''
270 INPUT ''+ '';B$
280 A$=A$+B$
290 GOTO 120
300 IF NWDS=0 OR NN=1 THEN 370
310 M$=LEFT$(A$,X(1)-1)
320 FOR J=1 TO NWDS
330 IF M$=N$(J) THEN A$=O$(J)+MID$(A$,LEN(N$(J))+1)
340 NEXT J
350 NN=1
360 GOTO 120
370 FLAG=0:B$=''NIL''
380 IF LEFT$(A$,5)=''CAR ('' THEN FLAG=1
390 IF LEFT$(A$,5)=''CDR ('' THEN FLAG=2
400 IF LEFT$(A$,6)=''CONS ('' THEN FLAG=3
410 IF LEFT$(A$,6)=''ATOM ('' THEN FLAG=4
420 IF LEFT$(A$,4)=''EQ ('' THEN FLAG=5
430 IF LEFT$(A$,6)=''NULL ('' THEN FLAG=6
440 IF LEFT$(A$,8)=''MEMBER ('' THEN FLAG=7
450 IF LEFT$(A$,6)=''MEMQ (''THEN FLAG=8
460 IF LEFT$(A$,8)=''APPEND (''THEN FLAG=9
470 IF LEFT$(A$,9)=''REVERSE ('' THEN FLAG=10
480 IF LEFT$(A$,7)=''EQUAL ('' THEN FLAG=11
490 IF LEFT$(A$,6)=''LIST ('' THEN FLAG=12
500 IF LEFT$(A$,8)=''DEFINE ('' THEN FLAG=13
510 IF LEFT$(A$,6)=''ADD1 ('' THEN FLAG=14
520 IF LEFT$(A$,7)=''SUB1 ('' THEN FLAG=15
530 IF LEFT$(A$,7)=''ZEROP ('' THEN FLAG=16
540 IF LEFT$(A$,12)=''DIFFERENCE ('' THEN FLAG=17
550 IF LEFT$(A$,6)=''EXPT ('' THEN FLAG=18
560 IF LEFT$(A$,5)=''MAX ('' THEN FLAG=19
570 IF LEFT$(A$,5)=''MIN ('' THEN FLAG=20
580 IF LEFT$(A$,6)=''PLUS ('' THEN FLAG=21
590 IF LEFT$(A$,7)=''MINUS ('' THEN FLAG=22
600 IF LEFT$(A$,10)=''QUOTIENT ('' THEN FLAG=23
610 IF LEFT$(A$,7)=''RECIP ('' THEN FLAG=24
620 IF LEFT$(A$,11)=''REMAINDER ('' THEN FLAG=25
630 IF LEFT$(A$,7)=''TIMES ('' THEN FLAG=26
640 IF LEFT$(A$,10)=''GREATERP ('' THEN FLAG=27
650 IF LEFT$(A$,7)=''LESSP ('' THEN FLAG=28
660 IF LEFT$(A$,8)=''MINUSP ('' THEN FLAG=29
```

```
670 IF LEFT$(A$,9)=''NUMBERP ('' THEN FLAG=30
680 IF LEFT$(A$,6)=''ONEP ('' THEN FLAG=31
690 IF FLAG>13 THEN 720
695 IF FLAG=0 THEN 70:REM PALAVRA NÃO RECONHECIDA
700 ON FLAG GOSUB 840,890,970,1110,1200,1330,1380,1510,1700,
    1760,2000,2130,3300
710 GOTO 760
720 IF FLAG>24 THEN 750
730 ON (FLAG-13) GOSUB 2180,2230,2280,2350,2560,2600,2790,
    2820,2870,2920,2960, 740
740 GOTO 760
750 ON (FLAG-24) GOSUB 3030,3070,3120,3160,3200,3250,2280
760 IF FLAG<>0 THEN GOSUB 780
770 GOTO 70
780 REM ** RESPOSTA **
790 PRINT ''   VALUE IS...''
800 IF B$<>''( )'' THEN PRINT''   '';B$
810 IF B$=''( )'' THEN PRINT''   NIL''
820 RETURN
830 REM ********************
840 REM ** CAR **
850 IF S=2 THEN B$=MID$(A$,Z(2)+1,X(2)-Z(2))
860 IF S> THEN B$=MID$(A$,CFIRST,CSECND-CFIRST+1)
870 RETURN
880 REM ********************
890 REM ** CDR **
900 GOSUB 840
910 LB=LEN(B$)+7
920 B$=''(''+MID$(A$,LB,EDGE-1)
930 IF RIGHT$(B$,2)=''))'' THEN B$=LEFT$(B$,LEN(B$)-1)
940 IF MID$(B$,2,1)='' ''THEN B$=''(''+MID$(B$,3)
950 RETURN
960 REM ********************
970 REM ** CONS **
980 B$=MID$(A$,7,LEN(A$)-1)
990 J=0
1000 IF LEFT$(B$,1)=''('' THEN J=1
1010 J=J+1
1020 IF MID$(B$,J,1)=''('' THEN 1050
1030 IF J<LEN (B$) THEN 1010
1040 B$=''   >CONS ERROR<'':RETURN
1050 LB=LEN(B$)-1
1060 B$=''(''+LEFT(B$,J-1)+MID$(B$,J+1)
1070 B$=LEFT$(B$,LB)
1080 IF RIGHT$(B$,2='' )'' THEN B$=LEFT$(B$,LEN(B$)-2)+'')''
```

```
1090 RETURN
1100 ********************
1110 REM ** ATOM **
1120 A$=MID$(A$,7,LEN(A$)-1)
1130 J=0:B$=''NIL''
1140 J=J+1
1150 IF MID$(A$,J,1)='' '' OR MID$(A$,J,1)=''('' THEN RETURN
1160 IF J<LEN(A$) THEN 1140
1170 B$=''T''
1180 RETURN
1190 REM ********************
1200 REM ** EQ **
1210 E=5:GOSUB 40
1220 J=0
1230 J=J+1
1240 IF MID$(A$,J,1)='')'' THEN RETURN
1250 IF MID$(A$,J,1)='' ''THEN 1280
1260 IF J<LEN(A$) THEN 1230
1270 RETURN
1280 C$=LEFT$(A$,J-1)
1290 A$=MID$(A$,J+1)
1300 IF C$=A$ THEN B$=''T''
1310 RETURN
1320 REM ********************
1330 REM ** NULL **
1340 IF A$=''NULL ( )'' THEN B$='' ILLEGAL - NULL NEEDS
     ARGUMENT''
1350 IF A$=''NULL (( ))'' THEN B$=''T''
1360 RETURN
1370 REM ********************
1380 REM ** MEMBER **
1390 C$=MID$(A$,9)
1400 J=1
1410 J=J+1
1420 IF MID$(C$,J,1)='')'' OR MID$(C$,J,1)=''(''THEN
     D$=LEFT$(C$,J) :GOTO 1450
1430 IF J<LEN(C$) THEN 1410
1440 RETURN
1450 J=LEN(D$)
1460 J=J+1
1470 IF MID$(C$,J,LEN(D$))=D$ THEN C$=LEFT$(C$,LEN(C$)-
     1):GOTO 1630
1480 IF J<LEN(C$) THEN 1460
1490 RETURN
1500 REM ********************
```

```
1510 REM ** MEMQ **
1520 C$=MID$(A$,7)
1530 J=0
1540 J=J+1
1550 IF MID$(C$,J,1)='' '' THEN 1580
1560 IF J<LEN(A$) THEN 1540
1570 RETURN
1580 D$=LEFT$(C$,J)
1590 C$=MID(C$,J+2)
1600 C$=LEFT$(C$,LEN(C$)-2)+'' ''
1610 J=0
1620 J=J+1
1630 IF MID$(C$,J,LEN(D$))=D$ THEN B$=''(''+MID$(C$,J):GOTO
     1660
1640 IF J<LEN(C$) THEN 1620
1650 RETURN
1660 B$=LEFT$(B$,LEN(B$)-1)+'')''
1670 IF RIGHT$(B$,3)=''))'' THEN B$=LEFT$(B$,LEN(B$)-
     1):GOTO 1670
1680 RETURN
1690 REM ********************
1700 REM ** APPEND **
1710 B$=MID$(A$,9)
1720 B$=LEFT$(B$,Y(1)-9+'' ''+MID$(B$,Z(3)-7)
1730 B$=LEFT$(B$,LEN(B$)-1)
1740 RETURN
1750 REM *******************
1760 REM ** REVERSE **
1770 B$='' ''
1780 A$=MID$(A$,11):A$=LEFT$(A$,LEN(A$))
1790 CT=0
1800 J=0
1810 J=J+1:IF J>LEN(A$) THEN 1920
1820 IF MID$(A$,J,1)='' '' THEN 1580
1830 IF MID$(A$,J,1)=''('' THEN 1860
1840 GOTO 1810
1850 CT=CT+1:G$(CT)=LEFT$(A$,J-1):A$=MID$(A$,J+1):GOTO 1800
1860 J=J+1:IF MID$(A$,J,2)=''))'' THEN 1980
1870 IF MID$(A$,J,1)='')'' THEN 1910
1880 IF J=LEN(A$) THEN 1900
1890 GOTO 1860
1900 CT=CT+1:G$(CT)=A$+.'')'':GOTO 1930
1910 CT= CT+1:G$(CT)=LEFT$(A$,J):A$=MID$(A$,J+=1):GOTO 1800
1920 CT=CT+1:G$(CT)=A$
1930 FOR M=CT TO 1 STEP -1
```

```
1940 B$=B$+G$(M):IF M>1 THEN B$=B$+'' ''
1950 NEXT M
1960 B$=''(''+B$+'')''
1970 RETURN
1980 CT=CT+1:G$(CT)=LEFT$(A$,J+1):A$=MID$(A$,J+2):GOTO 1800
1990 REM ******************
2000 REM ** EQUAL **
2010 E=8:GOSUB 40
2020 M=ASC(A$):IF M>47 AND M<58 THEN 2370
2030 J=0
2040 J =J+1
2050 IF MID$(A$,J,2)='') '' THEN J=J+1:GOTO 1280
2060 IF MID$(A$,J,3)='')))'' THEN 2100
2070 IF MID$(A$,J,2)=''))'' THEN 2110
2080 IF J<LEN(A$) THEN 2040
2090 RETURN
2100 C$=LEFT$(A$,J+2):A$=MID$(A$,J+4):GOTO 1300
2110 C$=LEFT$(A$,J+1):A$=MID$(A$,J+3):GOTO 1300
2120 REM ********************
2130 REM ** LIST **
2140 E=7:GOSUB 40
2150 B$=''(''+A$+='')''
2160 RETURN
2170 REM ********************
2180 REM ** ADD1 **
2190 E=7:GOSUB 40
2200 B$=STR$(VAL(A$)+1)
2210 RETURN
2220 REM ********************
2230 REM ** SUB1 **
2240 E=7:GOSUB 40
2250 B$=STR$(VAL(A$)-1)
2260 RETURN
2270 REM *******************
2280 REM ** ZEROP **
2290 IF FLAG=16 THEN E=8
2300 IF FLAG=31 THEN E=7
2310 GOSUB 40
2320 IF (A$=''0'' AND FLAG=16) OR (A$=''1'' AND FLAG=31)
     THEN B$=''T''
2330 RETURN
2340 REM ********************
2350 REM ** DOIS ARGUMENTOS **
2360 E=13:GOSUB 40
2370 J=0
```

```
2380 J=J+1
2390 IF MID$(A$,J,1)='' '' THEN 2420
2400 IF J<LEN(A$) THEN 2380
2410 B$='' * ERROR - ONLY ONE ARGUMENT *'':RETURN
2420 P=VAL(LEFT$(A$,J-1))
2430 Q=VAL(MID$(A$,J+1))
2440 IF FLAG=17 THEN B$=STR$(P-Q):RETURN
2450 IF FLAG=23 OR FLAG=25 THEN B=P/Q
2460 IF FLAG=25 THEN B=INT(.5+Q*(B-INT(B))*1000)/1000
2470 IF FLAG=18 THEN B=P^Q
2480 IF FLAG=11 AND P=Q THEN B$=''T''
2490 IF FLAG=27 AND P>Q THEN B$=''T''
2500 IF FLAG=28 AND P<Q THEN B$=''T''
2510 IF FLAG=32 THEN B=P-Q
2520 IF FLAG=11 OR FLAG>26 THEN RETURN
2530 B$=STR$(B)
2540 RETURN
2550 REM ********************
2560 REM ** EXPT **
2570 E=7:GOSUB 40
2580 GOTO 2370
2590 REM ********************
2600 REM ** MAX MIN PLUS TIMES **
2610 F$=LEFT$(A$,3):A$=MID$(A$,6)
2620 CT=0:FLAG=0
2630 IF F$=''TIMES'' THEN CT=1
2640 J=0
2650 J=J+1
2660 IF MID$(A$,J,1)='' '' THEN 2690
2670 IF J<LEN(A$) THEN 2650
2680 IF J=LEN (A$) THEN FLAG=1
2690 P=VAL(LEFT$(A$,J-1)):IF FLAG=0 THEN A$=MID$(A$,J+1)
2700 IF F$<>''PLUS'' AND CT=0 THEN CT=P
2710 IF F$=''MAX'' AND P>CT THEN CT=P
2720 IF F$=''MIN'' AND P<CT THEN CT=P
2730 IF F$=''PLUS'' THEN CT=CT+P
2740 IF F$=''TIMES'' THEN CT=CT*P
2750 IF FLAG=0 THEN 2640
2760 B$=STR$(CT)
2770 RETURN
2780 REM ********************
2790 REM ** MIN **
2800 GOTO 2610
2810 REM ********************
2820 REM ** PLUS **
```

```
2830 F$=''PLUS''
2840 A$=MID$(A$,7)
2850 GOTO 2620
2860 REM ********************
2870 REM ** MINUS **
2880 E=8:GOSUB 40
2890 B$=STR$(-VAL(A$))
2900 RETURN
2910 REM ********************
2920 REM ** QUOCIENTE **
2930 E=11:GOSUB 40
2940 GOTO 2370
2950 REM ********************
2960 REM ** RECIP **
2970 E=8:GOSUB 40
2980 IF A$=''0'' THEN B$='' DIVISION BY ZERO ILLEGAL'':RETURN
2990 B=1/(VAL(A$))
3000 B$=STR$(B)
3010 RETURN
3020 REM ********************
3030 REM ** RESTO **
3040 E=12:GOSUB 40
3050 GOTO 2370
3060 REM ********************
3070 REM ** TIMES **
3080 F$=''TIMES''
3090 A$=MID$(A$,8)
3100 GOTO 2620
3110 REM ********************
3120 REM ** GREATERP **
3130 E=11:GOSUB 40
3140 GOTO 3270
3150 REM ********************
3160 REM ** LESSP **
3170 E=8:GOSUB 40
3180 GOTO 2370
3190 REM ********************
3200 REM **MINUSP **
3210 E=9:GOSUB 40
3220 IF VAL(A$)<0 THEN B$=''T''
3230 RETURN
3240 REM ********************
3250 REM ** NUMBERP **
3260 A$=MID$(A$,10)
3270 IF ASC(A$)>44 AND ASC(A$)<58 THEN B$=''T''
```

```
3280 RETURN
3290 REM ********************
3300 REM ** DEFINIR **
3310 A$=MID$(A$,9)
3320 F$=LEFT$(A$,X(2)-9)
3330 G$=MID$(A$,X(4)-6)
3340 J=0
3350 J=J+1
3360 IF MID$(G$,J,1)='' '' THEN 3390
3370 IF J<LEN(G$) THEN 3350
3380 B='' DEFINE ERROR'':RETURN
3390 G$=LEFT$(G$,J-1)
3400 NWDS=NWDS+1
3410 O$(NWDS)=G$:N$(NWDS)=F$
3420 B$=F$
3430 RETURN
3440 REM ********************
3450 REM INICIALIZAR
3460 CLS
3470 DIM G$(20),O$(20),N$(20),X(12),Y(12),Z(12)
3480 NWDS=0:REM CONTAGEM DE PALAVRAS NOVAS DO UTILIZADOR
3490 GOTO 70
```

APÊNDICES

A

O TEOREMA DE BAYES E AS PROBABILIDADES

O Reverendo Thomas Bayes, que viveu entre 1702 e 1781, era um matemático notável. Os sistemas periciais nem sequer eram ainda sonhados (e se o fossem, seriam provavelmente atribuídos ao diabo). Contudo, o seu trabalho tem aplicação em qualquer campo (como estes sistemas) em que a probabilidade dos acontecimentos deva ser modificada à medida que se obtem informação adicional.

É provável que Bayes não ficasse satisfeito com esta aplicação da sua obra. Começou a desenvolver o seu teorema ao apresentar a admirável hipótese de a existência do Todo Poderoso poder ser demonstrada examinando as belezas matemáticas presentes no mundo por Ele criado. Provarei, disse, que o fim principal da divina providência é a felicidade das Suas criaturas, e fá-lo-ei usando as matemáticas.

Infelizmente, quanto mais avançou nos seus estudos, maior foi o seu alarme ao aperceber-se das implicações das suas descobertas. Decidiu-se finalmente a interromper a sua obra, e a não a publicar.

No entanto, a obra que fez era sólida e constitui o centro da moderna teoria da tomada de decisões. O seu teorema dá-nos uma forma matemática correcta de avaliação de novas informações, e do seu uso para modificar as estimativas anteriores — baseadas em dados limitados. Permite-nos actuar a partir de conhecimentos parciais, avaliando e revendo as nossas decisões à medida que aumentam o número de dados.

Para compreender o Teorema de Bayes e qual o seu valor para o desenvolvimento de sistemas periciais, temos de conhecer alguma coisa de probabilidades.

A probabilidade de um acontecimento ocorrer é o número de resultados com êxito dividido pelo número de resultados possíveis. Isto é, a probabilidade de uma moeda cair mostrando «caras» é 1 (o número de casos em que se obtém o resultado desejado) dividido por dois (o número de resultados possíveis).

Transformando isto numa equação, onde se usa P para indicar a probabilidade de um dado acontecimento, escrevemos:

$$P(\text{resultado}) = \frac{\text{número de resultados com êxito}}{\text{número de resultados possíveis}}$$

No nosso exemplo de lançamento de moedas, poderíamos escrever:

$$P(caras) = \frac{1}{1 + 1} = \frac{1}{2} = 0{,}5$$

Se se está a examinar mais do que um acontecimento (em vez do caso da moeda, em que apenas nos interessa um lançamento da moeda), deve-se separar os acontecimentos que se excluem reciprocamente (se está a chover, não pode estar a não chover) dos que não se encontram nessas circunstâncias (pode estar frio *e* nevoeiro).

Acontecimentos que se excluem mutuamente

A probabilidade destes acontecimentos é a probabilidade de ocorrer o resultado *x ou* o resultado *y*. Estas duas probabilidades somam-se, como se segue:

P(resultado X OU resultado Y) = P(resultado X) + P(resultado Y)

Imaginemos que estamos a lançar um dado. Queremos saber primeiro qual a probabilidade de mostrar um três:

$$P(três) = \frac{1}{6}$$

A probabilidade de mostrar um cinco, P(cinco), é exactamente igual, uma em seis. O dado não pode cair mostrando simultaneamente um três e um cinco, pelo que os acontecimentos excluem-se mutuamente. Isto significa que a probabilidade de o dado cair mostrando um três ou um cinco pode ser expressa do seguinte modo:

P(três OU cinco) = P(três) + P(cinco) = 1/6 + 1/6 = 2/6 = 1/3

Por outras palavras, há uma possibilidade em três de obtermos um três ou um cinco num único lançamento de dado.

Por outro lado, quanto atiramos o dado, pode mostrar um três ou um cinco, ou não mostrar nenhum destes valores. Qual será a probabilidade de cair sem mostrar três ou cinco? Obviamente, esta probabilidade é 2/3, isto é, 1–1/3. Quando somamos a probabilidade de um

acontecimento ocorrer com a probabilidade de não ocorrer, a soma deve ser sempre um:

```
P(acontecimento) + P(não acontecimento) = 1
```

... o que equivale a:

```
P(não acontecimento) = 1-P(acontecimento)
```

Acontecimentos que não se excluem mutuamente

Imaginemos agora que temos uma chave que serve para uma de seis caixas que se encontram à nossa frente sobre uma mesa. Não sabemos porém qual é a caixa que lhe corresponde. Conhecemos por outro lado o que se encontra no interior das caixas: um bloco negro, uma esfera negra, um bloco verde, uma esfera vermelha, uma escova para os dentes negra, e um balão azul. Experimentamos a chave em cada uma das caixas, até abrir uma delas. Qual é a probabilidade de a caixa aberta conter um objecto negro ou uma esfera? Obviamente, estes acontecimentos não se excluem mutuamente, dado que existe um objecto (a esfera negra) que satisfaz os dois critérios.

No entanto, a abertura de uma caixa que contém uma bola não significa necessariamente que esta bola seja negra. Mas como há a possibilidade de a abertura de uma caixa satisfazer ambas as condições, necessitamos de deduzir a probabilidade de uma condição ser satisfeita da possibilidade de ambas o serem.

A probabilidade de obtermos um objecto negro é 3/6, e a de obtermos uma bola é 2/6. A equação para selecção de uma bola ou um objecto negro (isto é, a probabilidade de um ou mais acontecimentos que não são mutuamente exclusivos) é:

```
P(negro OU bola) = P(negro) + P(bola) - P(negro E bola)
```

Em palavras, esta equação significa: a probabilidade de escolher um objecto negro ou uma esfera é igual à probabilidade de escolher um objecto negro mais a probabilidade de escolher uma bola, menos a probabilidade de escolher um objecto negro que seja também uma bola.

```
P(negro OU bola) = 3/6 + 2/6 - 1/6
                 = 5/6 - 1/6
                 = 4/6
                 = 2/3
```

Independência estatística

Talvez o leitor se interrogue sobre quais são as relações entre isto e o Reverendo Bayes. Há de facto uma relação, que será esclarecida adiante. O trabalho de Bayes não pode ser explicitado até termos tratado um pouco melhor as nossas probabilidades.

Se, quando batemos numa face com o nosso poderoso punho, a pessoa ofendida desenvolve um olho negro, dizemos que os dois acontecimentos são estatisticamente dependentes. A probabilidade de o segundo acontecimento ocorrer (o olho ficar negro) está directamente relacionada com a probabilidade de esmurrarmos alguém com suficiente força. No entanto, o simples facto de dois acontecimentos ocorrerem de seguida não significa que sejam estatisticamente dependentes. Lancemos uma moeda ao ar. Cai com o lado «caras» para cima. Lancemo-la novamente. A probabilidade de obtermos o mesmo resultado da segunda vez é completamente independente do primeiro lançamento.

Se chamarmos P(quatro) à probabilidade de obtermos um quatro quando lançamos um dado e P(seis) à probabilidade de obtermos um seis, a probabilidade de obtermos um quatro no primeiro lançamento e um seis no segundo pode ser expressa do seguinte modo (onde P(quatro & seis) indica a probabilidade de ocorrerem quatro e seis em sucessão):

```
P(quatro & seis) = P(quatro) X P(seis)
```

A probabilidade de obter um quatro é 1/6; e a probabilidade de obter um seis é igualmente de 1/6, pelo que a probabilidade de obter um quatro seguido de seis é 1/6 vezes 1/6, isto é, 1/36. A probabilidade de obtermos um quatro seguido de um seis e de um três será assim 1/6 vezes 1/6 vezes, ou 1/216. Sob a forma de uma equação, teremos:

```
p(quatro & seis & três) = P(quatro) × P(seis) × P(três)
```

A possibilidade de isto *não* ocorrer (ou seja, de não obtermos um quatro seguido de um seis e de um três) é igual à unidade menos a probabilidade de os três acontecimentos ocorrerem pela ordem indicada:

```
P(NÃO (quatro & seis & três)) =
          1 - (P(quatro) × P(seis) × P(três))
```

Esta forma de determinar a probabilidade de um acontecimento não ocorrer é verdadeira para qualquer situação; subtrai-se simplesmente da unidade a probabilidade de um acontecimento ocorrer. Se a probabilidade de obter um seis quando lançamos um dado, P(seis), for 1/6, a probabilidade de não obter um seis é 1 – 1/6, isto é, 5/6.

Probabilidade condicional

A probabilidade condicional refere-se à probabilidade de o acontecimento Y ocorrer depois de o acontecimento X ter ocorrido. Escreve-se sob a forma P(Y/X). Se os acontecimentos são estatisticamente independentes, a probabilidade de o acontecimento Y ocorrer depois de se observar o acontecimento X é (talvez surpreendentemente) apenas a probabilidade de ocorrer isolado, P(Y).

Porque acontece isto? Se lançarmos um dado, a probabilidade de produzir um seis é 1/6. Se o lançarmos de novo, a probabilidade de obtermos ainda um seis é novamente de 1/6. Um lançamento de dado não influencia o que acontece nos lançamentos seguinte.

Note-se que neste caso, o da independência estatística, estamos a perguntar qual é a probabilidade de o acontecimento Y ocorrer sabendo, antes, já ocorreu o acontecimento X; não estamos a perguntar qual é a probabilidade de ocorrerem os acontecimentos X E Y. Tendo já acontecido X (obtivémos seis num lançamento), qual é a probabilidade do acontecimento Y (obter um seis num lançamento)?

Dependência estatística

A questão torna-se um pouco mais complicada quando a probabilidade do segundo acontecimento está relacionada com a probabilidade do primeiro.

Probabilidade condicional

Imaginemos uma situação em que possuímos uma chave que abre uma de nove caixas que se encontram à nossa frente. Cinco destas contêm livros escritos por Tim Hartnell, duas contêm livros escritos pelo Dr. Rodnay Zaks, e as outras duas contêm livros escritos por Grace Murray Hopper (o Comodoro Hopper, quando este livro foi escrito, era o oficial feminino mais antigo na marinha americana e o seu oficial de idade mais avançada, apesar de ter tentado reformar-se várias vezes; Hopper concebeu a linguagem de computadores COBOL).

A probabilidade de a caixa a que a chave corresponde conter qualquer dos livros é 1/9. Existem nove livros e qualquer deles tem a mesma probabilidade de se encontrar na caixa aberta pela chave. No entanto, suponhamos que a caixa em questão contém um livro escrito por um autor masculino. A probabilidade de isto acontecer é 7/9. Qual é a probabilidade de ter sido escrito pelo Dr. Rodnay Zaks? Poderemos representá-la por P(Z:M), a probabilidade de o livro ser de Zaks, P(K), sabendo-se que se trata de um autor masculino, ao que corresponde a probabilidade P(M).

Sabemos que o livro é de um autor masculino. Para determinarmos a probabilidade de ser uma obra de Zaks, ignoramos os livros escritos por Hopper, dado que estes não podem estar envolvidos nesta situação. Sabemos que existem sete livros escritos por homens, e que dois deles o foram por Zaks. Para determinarmos a probabilidade que corresponde a Hartnell e a Zaks, dividimos o número de livros de cada autor pelo número total de livros de autores masculinos:

```
P(Z:M) = 2/7
P(H:M) = 5/7
```

A soma destas duas probabilidades é um. A probabilidade de um livro ser escrito por Zaks, tratando-se de uma obra de um autor masculino, é 2/7; e a probabilidade de ser escrito por Hartnell nas mesmas circunstâncias é 5/7.

Há portanto uma maior probabilidade, sendo a obra de um autor masculino, de ter sido escrita por Hartnell. Para determinamos a probabilidade de o livro ser de Zaks, dado ter sido escrito por um homem, P(Z:M), dividimos a probabilidade de Zaks pela probabilidade de homem, P(M), onde P(M) é igual a P(Z) mais P(H):

$$P(Z:M) = \frac{P(Z)}{P(M)} = \frac{P(2/9)}{P(7/9)} = \frac{.2222}{.7777} = .286$$

Como verificação disto, podemos raciocinar que existem sete livros de autores masculinos. Se bem que o que temos é de um homem, há duas possibilidades (num total de sete) de ter sido escrito por Zaks. Portanto, se o nosso método de determinar P(Z:M) estiver correcto, deveria dar a mesma resposta 2/7 (que é a probabilidade de ser escolhido um livro de Zaks quando existem duas obras deste num total de sete). Verificamos que assim acontece.

A probabilidade condicional, portanto, quando é estatisticamente dependente, pode ser expressa sob a forma:

```
P(Y:X) = P(YX)/P(X)
```

Voltando a Bayes

Estamos finalmente em posição de apreciarmos o trabalho do reverendo Bayes. Dissémos no início deste apêndice que o Teorema de Bayes nos dava um modo de utilizar novas informações para modificar estimativas anteriores, baseadas em dados limitados.

Imaginemos que temos duas caixas, cada uma delas contendo 25 bolas de madeira. Numa delas (B1) existem 14 bolas negras e 11 vermelhas. Na segunda caixa (B2) existem 19 bolas negras e seis vermelhas. Escolhemos uma caixa ao acaso, metemos a mão dentro dela, e tiramos uma bola. É negra. Qual é a probabilidade de termos tirado a bola de B2?

A probabilidade tanto para B1 como para B2 é 1/2 (0,5). A probabilidade de tirar uma bola negra de B1 é 14/25 (0,56), e a de obter uma bola da mesma cor de B2 é 19/25 (0,76). A probabilidade de usar B1 e tirar uma bola negra é 0,5 vezes 0,56 (0,28), e a de utilizar B2 e obter uma bola negra é 0,5 vezes 0,76 (0,38). A probabilidade de obter uma bola negra em qualquer caso é a soma destas duas probabilidades, 0,28 mais 0,38 (0,66). Nestas condições, a probabilidade de termos obtido a bola negra em B1 é P(B1,negra)/P(negra), ou 0,28 dividido por 0,66, o que equivale a 0,424; e a probabilidade de a termos obtido em B2 é 1 – 0,424, ou 0,576 (como só podemos obter a bola em B1 ou em B2, a soma das probabilidades deve ser igual à unidade).

Que nos diz isto? Que significado têm os valores 0,424 ou 0,576? Antes de abrirmos uma caixa e tirarmos uma bola, diríamos que a probabilidade de a bola ser obtida em B1 ou B2 seria de 0,5; mas agora, depois de escolhermos apenas uma bola, podemos dizer que há uma maior probabilidade de esta ter saído de B2 do que de B1.

Coloquemos novamente a bola em B2. Tentemos «baralhar» as caixas. Escolhamos uma caixa ao acaso, e tiremos uma bola de dentro dela. Imaginemos que conseguimos tirar outra bola negra. Poderemos dizer, com confiança, que a tirámos outra vez da caixa negra? A probabilidade de as duas bolas terem saído da caixa B1 é 0,5 vezes 0,56 vezes 0,56 (0,129), e a probabilidade de as bolas provirem de B2 é 0,5 vezes 0,76 vezes 0,76 (0,289). Se somarmos estes valores, obteremos a probabilidade de tirarmos duas bolas negras seguidas (0,159 mais 0,289, ou seja, 0,448).

Como determinamos agora a probabilidade de a segunda bola ser tirada de B2?

A probabilidade de tirarmos duas bolas negras de B1 é a de escolher B1 e tirar estas duas bolas (0,159) divididas pela probabilidade de obtermos duas bolas negras de seguida (0,448), o que é igual a 0,355. A probabilidade de obtermos duas bolas de seguida em B2 deve ser 1 –0,355 ou 0,645. Vejamos se de facto assim é. A probabilidade de escolher B2 e obter duas bolas negras de dentro dela (0,289) dividida pela probabilidade de obter duas bolas negras de seguida (0,448) é com efeito 01645.

Que concluir daqui? Saberemos de facto algo mais do que sabíamos no início? Começámos todo este processo de caixas e bolas tendo apenas a informação de que havia uma possibilidade em duas (0,5) de escolhermos B1 ou B2. Depois de tirarmos uma única bola, que se ve-

rificou ser negra, foi-nos possível dizer que a probabilidade de ter saído de B1 era de 0,424 e a de provir de B2 era 0,576.

Colocámos de novo a bola na caixa, «baralhámos» as caixas, e tirámos uma nova bola. Era novamente negra. Repetimos os nossos cálculos, e descobrimos que a probabilidade de a bola ter sido tirada de B1 era de 0,355, sendo de 0,645 a probabilidade de a termos obtido na caixa B2. Podemos portanto dizer que, se obtemos duas bolas de seguida (colocando de novo a primeira bola na caixa depois de a termos tirado) e ambas tiverem a cor negra, a probabilidade de ambas terem saído de B2 é 0,645.

B

BASE DE DADOS

A capacidade dos sistemas periciais no futuro dever-se-á à eficiência e inteligência do motor de inferência e à qualidade da base de conhecimentos a que o sistema pode aceder.

A informação que a base de conhecimentos pode conter terá, a curto prazo, uma natureza essencialmente textual, dado que os computadores actuais são bastante melhores a trabalhar os símbolos que representam texto do que símbolos mais complexos como os usados, por exemplo, para codificar uma imagem a cores animada. Actualmente, cerca de 55% do material tratado por uma empresa típica é texto, juntamente com 30% de dados e cerca de 15% de informação de imagem.

Quando se constrói uma base de dados para ser acedida por um sistema pericial, é necessário considerar três coisas. São as seguintes:

— O custo do armazenamento da informação
— A velocidade da comunicação da informação
— A qualidade da informação guardada.

As grandes dimensões da informação a manipular deve ser igualmente tida em conta.

O custo do armazenamento de dados

O custo do armazenamento de informação diminuíu dramaticamente nos últimos trinta anos, como a tabela seguinte — que mostra o custo grosseiro da manutenção de uma folha de papel almaço cheia de informação — ilustra de forma convicente. Os valores são indicados em dólares.

ANO	EM MEMÓRIA CENTRAL	EM PERIFÉRICOS ON-LINE
1950	$225 000 000.00	—
1960	$70 000.00	$12 000.00
1970	$11 500.00	$2 500.00
1975	$2 750.00	$250.00
1980	$275.00	$20.00
1983	$120.00	$15.00
1985 (estimativa)	$12.00	$0.75
1990 (estimativa)	$2.00	$0.12

Estes dois últimos serão provavelmente pessimistas. Em 1987, os suportes de armazenamento magnético constituirão de facto o método mais barato de guardar texto, e oferecerão evidentemente os benefícios adicionais da acessibilidade imediata, procura rápida de informação e rapidez de comunicação. Dois anos depois, o armazenamento electrónico constituirá a forma mais barata de guardar informação de imagem, permitindo armazenar informação de cor e animação.

As características dos dados no interior de bases de dados

Quais são as características das grandes quantidades de dados, como as guardadas em bases de conhecimento? Podemos dividir estes dados por três níveis. O primeiro é o dos dados puros, um conjunto de dados não relacionados entre si. O segundo nível é o do conhecimento estruturado, guardado numa forma que pode ser trabalhada. Esta estrutura é vital se se pretende dar algum significado à informação. O terceiro, e que em muitos casos será o nível mais importante, é o do conhecimento «associado», em que não se guarda apenas informação. Além desta, a base de conhecimentos contém as relações existentes entre os dados.

As decisões mais importantes que tomaremos na próxima década terão a ver com a forma como o conhecimento, e as suas relações, será guardado. Existem muitas formas de guardar bases de conhecimentos, e a informação sobre as relações existentes entre os elementos dessas bases. E o tipo de decisões que tomaremos tendo em conta a informação contida nas bases levar-nos-ão a usar estas formas de organização talvez para sempre. Um problema actual relacionado com isto é a enorme grandeza de tarefa de conversão das velhas bases de dados (guardadas por exemplo em ficheiros de papel) para formas electrónicas.

Em Inglaterra, por exemplo, o Departamento de Saúde e Segurança Social enfrenta neste momento dois problemas. Dispõe de cerca de

26,5 milhões de nomes em ficheiros activos, relacionados por localização geográfica e outros factores relevantes, e uma maciça «base de regras» contendo informações como o modo de execução de pagamentos, a quem são feitos e em que circunstâncias. De acordo com o Departamento citado, a tecnologia actual é incapaz de tratar a base de regras. No entanto, os métodos antiquados já não são apropriados, pelo que mesmo antes de existir a tecnologia necessária é forçoso tomar decisões quanto à forma como a informação será tratada de tal modo que, quando a tecnologia estiver disponível, ela não seja travada por estar reunida de forma primitiva. Isto indica o tipo de problema enfrentado pelas formas actuais de organizar o conhecimento.

Como é tratada a informação

O modo como a informação é actualmente tratada é formada por três componentes. O principal, neste momento, é a base de dados, seguida do trabalho aplicado em sistemas periciais. O terceiro componente, que discutirei dentro em pouco, está menos bem definido, mas constitui mesmo assim uma parte importante dos modos de que dispomos para guardar informação.

As bases de dados começaram por ser formadas por «arrays» bidimensionais estruturados, bastante simples, e daí evoluiram para aquilo a que se chama «bases de dados relacionais». O principal problema de uma base de dados é que é necessário saber de avanço a forma como a informação será organizada. Isto, como é fácil compreender, pode limitar severamente a eficácia e flexibilidade de uma base de dados.

O segundo componente da forma como os conhecimentos são actualmente tratatados é a construção de bases periciais para tratamento por motores de inferência.

O terceiro componente pode talvez ser referenciado pela expressão «tecnologia de transferência». Trata-se da capacidade de obter aptidões numa área e aplicá-las noutras, ou juntamente com outras tecnologias (como o ensino, treino por computador ou o uso de «vídeo interactivo», onde os discos-«laser» poderão dentro de pouco ser o componente mais importante).

O disco-«laser» constitui obviamente uma resposta a um dos problemas actuais, as elevadas quantidades de informação que é necessário guardar. Actualmente, um disco-«laser» pode conter cerca de 55 000 imagens fixas, o que equivale a cerca de 700 megabytes. A sua produção custa menos de cinco dólares cada. Estes valores são extraordinários. Estão actualmente a ser desenvolvidos discos que podem ser escritos pelo utilizador, dado que por enquanto só podem ser usados em leitura. As estatísticas mostram já as potencialidades que

estes discos terão. O principal problema reside na gestão de todos estes dados. Ao contrário do que acontece com a maior parte dos dados textuais, as imagens não são geralmente classificáveis de forma clara. No entanto, as vastas potencialidades de referenciação do computador, acrescentadas à sua capacidade de tratar dados com a densidade correspondente à de um disco laser, mostra que os discos vídeo só poderiam evoluir, e ter algum sentido, quando usados juntamente com um computador para controlar e organizar a informação.

A qualidade da informação guardada nos bancos de dados dos computadores deve ser igualmente examinada. A maior parte da informação é de facto gerida por lógica matemática (usando operadores como +, –, *, < e >). As palavras são tratadas como números. Isto significa que têm para o computador um valor numérico sem qualquer significado. Ou seja, sendo a informação tratada como números o sistema não tem a mais pequena ideia daquilo com que trabalha.

Se pudermos produzir um sistema que utilize as palavras como tais, manipuladas por lógica, poderíamos aproximar-nos bastante de uma máquina «consciente».

Isto já está a acontecer, até certo ponto. A maior capacidade de processamento, e o menor custo das memórias, significa que se tornou já possível desenvolver sistemas baseados na lógica verbal (semântica). Este tipo de sistemas podem reconhecer relações não matemáticas entre as palavras, como por exemplo a associação que existe entre pares de entidades como pai e filho, grande e pequeno ou cidade e Nova Iorque, como o leitor já teve ocasião de compreender ao observar o material contido neste mesmo livro, em programas como o HASTE, o EASLE, o PROLOG-A e o SSLISP, precisamente orientados para um relacionamento semântico das palavras.

C

REGRAS DA LÓGICA «DIFUSA»

A lógica «difusa», usada em programas como RITA, utiliza os operadores E, OU e NEGAÇÃO (AND, OR, NOT):

NOT: dadas duas condições opostas, a probabilidade de um estado é (1–probabilidade) do estado oposto

AND: Este assume o menor de dois (ou mais) valores, de tal modo que se um é 0,3 e outro 0,5, o uso de AND produz 0,3

OR: Este assume o maior de dois ou mais valores, de tal modo que se um é 0,3 e outro 0,5,. o uso de OR produz 0,5.

Note-se que este modo de determinar valores numa situação AND ou OR é em grande medida tradicional. Algumas pessoas argumentam que um AND deveria ser o *múltiplo* da probabilidade-um e probabilidade-dois. O uso do método tradicional parece dar resultados na prática, e dada a forma bastante empírica como é necessário avaliar a saída destes sistemas na maior parte dos casos, o facto de dar resultados é com efeito tudo o que interessa.

NOT p1 ------> 1–p1
p1 AND p2 ------> MIN(p1,p2)
p1 OR p2 ------> MAX(p1,p2)

D

DADOS METEOROLÓGICOS

São estes os dados básicos utilizados na secção de previsão do tempo de RITA. Como se pode ver, não foram usados todos os valores. Talvez o leitor esteja interessado em experimentar o sistema, utilizando a informação a que não dei uso no meu exemplo.

DATA	VALOR BAROM. ÀS 9 HORAS	TEMPERATURA		HUM. REL. (%) 15 H	VENTO, 15 H (KM/H)		RAJADAS MÁX. (KM/H)		HORAS DE SOL	EVAP. 24 HRS. (mm)	PRECIP. 24 HRS. (mm)
		MIN (°C)	MAX (°C)								
1	1011.6	11.0	25.5	31	SE	07	SO	28	13.5	3.4	0
2	1006.7	12.6	27.6	29	SO	11	NO	46	10.0	4.8	0
3	1012.4	10.8	16.5	52	S	13	SO	57	12.7	5.6	2.6
4	1014.7	9.1	15.9	74	O	15	O	72	0.0	4.0	1.6
5	1016.5	11.0	16.8	56	SSE	11	OSO	45	0.7	1.2	7.2
6	1018.1	13.0	17.9	60	S	07	SSO	26	1.2	4.2	0
7	1017.9	12.8	23.1	51	S	13	S	43	11.0	2.4	0
8	1016.7	9.9	22.8	38	SE	19	ESE	50	13.6	3.8	0
9	1013.9	11.6	24.2	38	SE	09	SE	45	13.7	6.0	0
10	1010.8	10.5	27.7	22	E	06	SE	45	11.8	5.6	0
11	999.0	15.9	21.3	60	Nulo		NO	50	4.5	6.2	2.6
12	1002.6	12.2	23.5	30	O	26	ONO	63	9.8	3.4	2.6
13	1005.9	13.0	17.9	86	NNO	06	O	30	2.6	4.0	0
14	1000.0	12.5	24.2	84	ENE	04	O	61	3.2	2.4	12.4
15	1003.0	12.5	20.0	46	S	11	OSO	46	10.4	1.2	7.4
16	1009.8	11.4	23.7	43	SE	11	S	41	12.9	5.8	0.4
17	1011.1	12.1	27.0	31	SSE	07	N	26	13.5	3.2	0
18	1002.0	16.8	24.6	36	OSO	26	OSO	67	10.0	6.8	0.2
19	1003.7	12.1	25.0	38	NO	19	O	67	7.6	7.0	0
20	1000.1	11.3	20.0	40	OSO	26	O	72	11.1	4.4	0.8
21	1010.1	10.4	22.7	37	O	11	OSO	43	9.1	2.8	1.2
22	1013.6	14.2	22.6	40	SSE	07	OSO	35	8.7	5.4	0
23	1011.8	11.4	29.2	32	SSE	06	S	43	13.4	5.6	0
24	1009.8	13.7	27.8	58	S	06	SSO	52	3.7	5.4	0
25	994.2	16.5	26.5	56	OSO	19	OSO	59	2.0	5.0	1.8
26	1012.0	12.8	18.9	66	SO	19	S	52	4.3	3.0	3.2
27	1017.6	12.2	23.2	41	ESE	13	E	41	14.1	6.6	2.0
28	1017.5	12.0	24.2	40	SE	11	ESE	48	13.4	5.4	0
29	1014.4	12.6	28.1	46	E	11	E	35	12.8	4.6	0
30	1009.6	16.0	30.3	37	S	15	S	41	13.1	5.6	0
31	1007.1	16.4	25.6	46	E	04	S	52	9.2	7.2	0
MÉDIA	1009.5	12.6	23.4	47					9.2	142.0	46.0
MÉDIA A LONGO PRAZO	1013.2	12.7	24.1	49					8.2	195.3	57.2

E
REFERÊNCIAS

CLARK, K. L., e F. G. McCabe, «Micro-Prolog: Programming in Logic», Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

BUCHANAN, B. e E. Fiegenbaum, «Dendral and Meta-Dendral: Their Aplications Dimension», em *Readings in Artificial Intelligence,* organizado por B. L. Weber e N. J. Nilsson, Tioga Publishing, Palo Alto, Califórnia, 1981.

BUCHANAN, B., e E. Shortliffe, «Rule-Based Expert Sistems», Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

DUDA, R., J. Gaschni e P. Hart, «Model Design in the PROSPECTOR Consultant System for Mineral Exploration», em *Readings in Artificial Intelligence,* B. L. Webber e N. J. Nilsson, organizadores, Tioga Publishing, Palo Alto, California, 1981.

FIEGENBAUM, E.A. e P. McCorduck, «The Fifth Generation», Addison-Wesley Publishing Company, Reading, Massachussetts, 1983.

HARTNELL, T., «Exploring Artificial Intelligence on your Microcomputer», Interface Publications, Londres, Inglaterra, 1984.

McCORDUCK, P., «Machines Who Think», W. H. Freeman & Co., San Francisco, 1979.

RAPHAEL, B., «The Thinking Computer», W. H. Freeman & Company, San Francisco, 1976.

RICH, E., «Artificial Intelligence». McGraw-Hill, Nova Iorque, 1983.

SHORTLIFFE, E., «Consultation Systems for Physicians», em *Readings in Artificial Intelligence,* B. L. Webber e N. J. Nilsson (organizadores), Tioga Publishing, Palo Alto, Califórnia, 1981.

WEBBER, B. L. e N. J. Nilsson, «Readings in Artificial Intelligence», Tioga Publishing, Palo Alto, California, 1981.

WHITE, D. e Shaw, W.P., «A Modern Introduction to Chemistry», Pergamon Press, Elsmford, Nova Iorque, 1980.

ZADEH, L.A, «A Theory of Approximate Reasoning», *Machine Intelligence 9,* Hayes J. E., D. Michie e L. I. Mikulich (organizadores), Halsted Press, John Wiley & Sons, Nova Iorque, 1979.

F

OUTRAS LEITURAS

BODEN, M., «Artificial Intelligence and Natural Man», Harvester Press, Basic Books Inc., Nova Iorque, 1981.
JAMES, M., «Artificial Intelligence in BASIC», Newnes Technical Books, Butterworth & Company, Londres, Inglaterra, 1984.
NAYLOR, C. M., «Build your own Expert Sistem», Sigma Technical Press, Cheshire, Inglaterra, 1983.
PEARL, J. «Heuristics — Intelligence Search Strategies for Computer Problema Solving», Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.
TORRANCE, S.(organizador), «The Mind and the Machine», Halsted Press, John Wiley & Sons, Nova Iorque, 1984.
SIMONS, G. L., «Towards Fifth-Generation Computers», NCC Publications, Manchester, Inglaterra, 1984.
WINSTON, P. H., «Artificial Inteligence», Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.
WOS, L., R. Overbeek, E. Lusk e J. Boyle, «Automated Reasoning; Introduction and Applications», Prentice-Hall, Englewood Cliffs, Nova Jersey, 1984.

Criação de Compiladores

NICHOLLS, J. E., «The Structure and Design of Programming Languages», Addison-Wesley Publishing Company, Reading, Massachusetts, 1975.
ROHL, J. S., «An Introduction to Compiler Writing», American Elsevier, Nova Iorque, 1975.

LISP

SIKLOSSY, L., «Let's Talk LISP», Prentice-Hall,Englewood Cliffs, New Jersey, 1976
«Learning LISP», Prentice-Hall, Englewood Cliffs, New Jersey, 1984

PROLOG

McCABE, F. G., K. L. Clark e B. D. Steel, «Micro-PROLOG 3.1 Programme's Reference Manual», Logic Programming Associates, Londres, Inglaterra, 1984.

POUNTAIN, D., «Prolog on Microcomputers», em *Byte Magazine,* McGraw-Hill, Dezembro, 1984.

Interface Publications Artificial Intelligence Library
(todos os títulos por Tim Hartnell)

Exploring Artificial Intelligence on your Microcomputer — 1984
Exploring Artificial Intelligence on your Spectrum + and Spectrum — 1984
Exploring Artificial Intelligence on your BBC Micro — 1985
Exploring Artificial Intelligence on your Commodore 64 — 1985
Exploring Artificial Intelligence on your QL — 1985
Exploring Expert Systems on your Microcomputer — 1985

AGRADECIMENTOS

Gostaria de apresentar os meus agradecimentos a Richard Forsyth, de Warm Boot Ltd., Londres, pela lista de características de um sistema pericial apresentada no primeiro capítulo. Richard organizou um seminário sobre inteligência artificial e aprendizagem da máquina que frequentei em 1984, e neste seminário tomei notas que constituiram a base do apêndice sobre bases de dados e outros elementos de informação em diversos locais do livro. Ainda neste seminário, Phil Cox falou sobre o sistema micro-pericial que ajudou a desenvolver. De Phil obtive alguma da informação sobre suficiência lógica e necessidade lógica que aparece no capítulo três.

Agradeço também o auxílio de Nicholas McLaren, da Melbourne, e do programador e autor Ross Symons, a propósito da secção sobre o teorema de Bayes.

ÍNDICE

INTRODUÇÃO 9

1. O QUE É UM SISTEMA PERICIAL 11
   - Os principais ingredientes 12
   - A base de conhecimentos 13

2. SISTEMAS QUE FUNCIONAM 16
   - As experiências MYCIN em Stanford 16
   - O «Gang» MYCIN 17
   - As regras 17
   - EMYCIN 19
   - DENDRAL, um feiticeiro químico 21
   - Sistemas milionários 23

3. CRIAÇÃO DE SISTEMAS 24
   - Tipos de informação 25
   - Dois tipos de raciocínio 26

4. SISTEMAS BASEADOS EM REGRAS 28
   - Duas faces de cada regra 28

5. O DOUTOR JÁ CHEGOU 36

6. DESENVOLVENDO UM SISTEMA PERICIAL 43
   - Conheça RITA 44
   - Os componentes de um sistema pericial 44
   - Um sistema geral 45
   - Será um gato ou um cão? 46
   - Maior dificuldade 51

7. RACIOCÍNIO «DIFUSO» 56
   - Voltando ao mundo real 57
   - Londres 57

Qual o valor destes resultados? 59
Modificações 61
Dissecando RITA 61
Dois resultados 67

8. LISTAGEM COMPLETA DE RITA 69

9. O GABINETE DE METEOROLOGIA 73
A educação de RITA 75
Novas entradas 79

10. LÓGICA E PROGRAMAÇÃO 80
Linguagens declarativas 80
LISP 81
PROLOG 82
Micro-PROLOG 82

11. PENSANDO EM «HASTE» 84

12. UM SABOR A PROLOG 90
A via marciana 92
Famílias felizes 96
O jogo dos números 102

13. A LISTAGEM PROLOG-A 106

14. UMA VERSÃO DA LISP 115
As palavras 115
As funções básicas 116
CAR 116
CDR 117
CONS 117
ATOM 118
EQ 119
NULL 119
EASLE 119

15. SSLIPS 123
MEMBER 123
MEMQ 124
APPEND 124
REVERSE 124
EQUAL 125

LIST 125
Operações matemáticas 126
Trabalhando numericamente 128

16. DEFINIÇÃO DE FUNÇÕES EM LISP 132

17. LISTAGEM DA SSLIPS 135

LISTAGENS ESPECÍFICAS PARA VÁRIOS MICROCOMPUTADORES 145
COMMODORE 64 147
SPECTRUM/SPECTRUM + ... 182
MICROCOMPUTADOR BBC 216

APÊNDICES:

A. O TEOREMA DE BAYES E AS PROBABILIDADES 251
Acontecimentos que se excluem mutuamente 252
Acontecimentos que não se excluem mutuamente 253
Independência estatística 254
Probabilidade condicional 255
Dependência estatística 255
Probabilidade condicional 255
Voltando a Bayes 256

B. BASE DE DADOS 259
O custo do armazenamento de dados 259
As características dos dados no interior de bases de dados 260
Como é tratada a informação 261

C. REGRAS DA LÓGICA «DIFUSA» 263

D. DADOS METEOROLÓGICOS 264

E. REFERÊNCIAS 265

F. OUTRAS LEITURAS 266

Os sistemas periciais vieram pôr ao alcance de um vasto público conhecimentos especializados que permitem solucionar problemas de várias ordens, desde a detecção de doenças à reparação, por exemplo, de avarias mecânicas em automóveis. Também são utilizáveis no campo da química, ou da meteorologia, não ficando por aqui as aplicações destes sistemas ainda em fase de pesquisa. Neste livro são analisados os sistemas periciais mais vulgarizados, o MYCIN, o DENDRAL, o PROSPECTOR, o MECÂNICO (é apresentado um programa para este sistema). Mas o sistema RITA é especialmente escolhido pelo autor para apresentar a estrutura genérica de um sistema pericial, bem como o seu modo de funcionamento. Entretanto há também muita informação sobre as linguagens que estão a ser mais utilizadas neste domínio — LISP, Micro-PROLOG, PROLOG e a novíssima HASTE. Além de uma listagem completa para RITA, são dadas listagens específicas para os computadores Commodore 64, Spectrum e Spectrum + e o microcomputador BBC, e outras ainda de grande utilidade. Esta obra contém apêndices sobre probabilidades, bases de dados, regras de lógica «difusa», indicações para outras leituras e pistas de investigação que serão muito úteis ao leitor interessado em aprofundar os seus conhecimentos nesta área.

OS SISTEMAS PERICIAIS EM MICROCOMPUTADORES

Tim Hartnell

29